



TECHNISCHE UNIVERSITÄT MÜNCHEN
Lehrstuhl für Raumfahrttechnik

TECHNICAL UNIVERSITY OF MUNICH
Chair of Astronautics

Reliability Assessment and Reliability Prediction of CubeSats through System Level Testing and Reliability Growth Modelling

Dipl.-Ing. Univ. Martin Langer

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Prof. Phaedon-Stelios Koutsourelakis, Ph.D.

Prüfer der Dissertation: 1.) Prof. Dr. rer. nat. Dr. h.c. Ulrich Walter

2.) Prof. Dr.-Ing. Enrico Stoll (Technische Universität Braunschweig)

Die Dissertation wurde am 21.06.2018 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 01.10.2018 angenommen.

Preface

“Difficulties are just things to overcome after all.”

Ernest Shackleton

“Nothing is impossible in this world. Firm determination, it is said, can move heaven and earth.”

Yamamoto Tsunetomo



This thesis is dedicated to my dad and to all the hard-working students of the MOVE-II team.

This page intentionally left blank.

Danksagungen

MOVE-II und diese Dissertation wären ohne die Unterstützung zahlreicher Menschen während der letzten 5 ½ Jahre nicht möglich gewesen. Zunächst möchte ich mich bei Prof. Walter bedanken, der mir es ermöglicht hat, an den LRT zu kommen und mich hier zu entfalten. Ich bin sehr dankbar für die Freiheit aber auch die wertvollen Ratschläge, die ich als Doktorand von ihm empfangen konnte. Diesen Dank möchte ich gerne auf Martin Rott erweitern, der den LRT als „finanzielles Mastermind“ zusammenhält, und mit dem ich jederzeit gerne wieder ein Durianeis essen gehen werde. Mein Dank gilt auch Frau Lochner, die mir gemeinsam mit Martin half, so manche bürokratische 5 Meter-Hürde (mit Wassergraben) zu überwinden.

Bei Prof. Stoll möchte ich mich für die Zweitbetreuung dieser Arbeit und für zahlreiche interessante Gespräche in der Vergangenheit bedanken. I would also like to thank Marco Villa, Prof. Eberhard Gill, Jasper Bouwmeester, Bulent Altan, and Prof. Michael Swartwout: your advice but also your critical questions helped me a lot and improved this work. Den zahlreichen Teilnehmern meiner Umfrage und den vielen CubeSat Entwicklern, die Erfahrungen mit mir ausgetauscht haben, gebührt ein großer Dank und auch meine Anerkennung für die immer sehr offenen Gespräche über CubeSat-Fehler. Ohne die Förderung des Deutschen Zentrums für Luft- und Raumfahrt, und hier gilt mein Dank allen voran Herrn Christian Nitzschke, wäre weder MOVE-II noch diese Arbeit möglich gewesen. Absolventen der verschiedensten CubeSat Projekte in ganz Deutschland zeigen jeden Tag, welch ein wunderbares Raumfahrt-Ausbildungsprogramm hier existiert.

In den mehr als 5 Jahren meiner Beschäftigung am LRT gab es keinen Tag, an dem ich nicht gerne in die Arbeit kam. Hierfür bin ich sehr dankbar, und möchte allen Kollegen am LRT meine Wertschätzung ausdrücken. Die Diskussionen über die österreichische Sprache oder Fußballergebnisse haben mir die Zeit mit euch noch mehr versüßt, aber ich konnte auch in zahlreichen fachlichen Gesprächen mit euch eine Menge lernen. Mein besonderer Dank gilt an dieser Stelle Alex Höhn, der immer ein offenes Ohr für technische (oder sonstige) Fragen hatte, egal ob MOVE-II oder meine Dissertation das Thema war. Leider kann ich immer noch nicht ganz aufklären, ob nun CubeSats Elefanten oder doch eher Pflanzen sind. Claas und Jan, danke für eure vielen Ratschläge und Hilfe, und erst durch euch bin ich am LRT so richtig angekommen. Basti, Flo, Nic, Martin, David und Jonis – danke, dass ihr meine Arbeit weiterführt und mir jeden Tag zeigt, welchen Wert die Arbeit in MOVE-II tatsächlich hat. Dem Rest des Teams möchte ich, ohne namentlich alle nennen zu können, nochmal vielen Dank für die letzten 5 ½ Jahre sagen – es hat mich sehr gefreut. Bedauerlicherweise ist der Titelgewinn im Fußball-Lehrstuhlturnier eher in Ferne gerückt, dafür werden wir in Zukunft sicherlich bei Darts erfolgreicher sein.

Diese Arbeit wäre ohne die zahlreichen motivierten Studenten des MOVE-II Teams nicht möglich gewesen. Stellvertretend für alle am Projekt beteiligten Studenten möchte ich mich bei den besonders engagierten bedanken (in alphabetischer Reihenfolge): Alex, Basti, Basti, Daniel, David, Flo, Florian, Jonis, Karl, Katja, Kim, Laura, Lucas, Lucie, Martin, Michael, Nicolas, Rupert, Till, Thomas, Tejas - ohne euch würde es diesen Satelliten, und damit auch meine Arbeit, heute sicherlich nicht geben! Auch bin ich für die zahlreichen Experten dankbar, die uns in MOVE-II immer wieder wertvolle Ratschläge und Hilfe gaben. Auch hier wäre die Liste zu lang, um mich bei jedem einzeln zu bedanken. Herausragend war die Hilfe von: Rolf-Dieter Klein, Prof. Dieter Kranzlmüller, Markus Plattner, Martin Rutzinger und Carsten Trinitis - Herzlichen Dank!

Es gibt zahlreiche Freunde deren Zuspruch und Interesse mich immer motiviert haben, die aber auch immer ein offenes Ohr für meine Sorgen hatten. Auch hier ist die Liste zu lang, daher kann ich nur einigen stellvertretend an dieser Stelle danken: Claus, Stephan, Michi, Pedro, David, Christoph, Franz, Pez, Gernot, Sophia, Alex und Ines, vielen Dank! Ich danke Peter Stangl, dass er meine Führungsqualitäten erkannt und geweckt hat, als diese noch tief in mir schlummerten. Ohne ihn wäre eine Leitung eines 100-köpfigen Teams für mich niemals denkbar gewesen. Ich möchte mich des Weiteren bei allen bedanken, die (oft nächtelang) meine Arbeit korrekturgelesen haben, und dadurch ihre Qualität merklich gehoben haben: Alex, Flo, Jasmin, Susi – Danke, und Respekt vor eurem Durchhaltevermögen! Der größte Dank gebührt meiner Familie, die mich immer auf meinem Weg unterstützt hat. Ich möchte hier auch die Familie meiner Frau miteinschließen, die in den letzten Jahren auch zu meiner Familie geworden ist – ich danke euch!

Meinem Vater ist diese Arbeit gewidmet, gemeinsam mit meiner Mutter hat er von früh weg mein Interesse an der Technik geweckt, aber immer auch darauf geachtet, dass die Balance zu anderen Dingen in meinem Leben stimmt. Beiden gebührt der Dank für einen wesentlichen Teil und den Erfolg meines bisherigen Lebenswegs. Bei meiner Schwester und ihrem Verlobten möchte ich mich für ihre immerwährende Unterstützung und ihren Zuspruch in allen Lebenslagen bedanken. Onkel Erhard möchte ich für viele Gespräche voller Wissen und Lebenserfahrung danken.

Der größte Dank gilt meiner Frau Simona, die mein Leben erst zu dem gemacht hat, was es heute ist. Dein Rat, deine Hilfe, deine Ideen, deine Freundschaft, dein Humor, deine Offenheit, deine Ehrlichkeit und deine Liebe – ohne alldem wäre ich nie so weit gekommen.

Abstract

This dissertation describes a set of methods used to track, assess and predict the reliability of CubeSats through system level testing and reliability growth modelling. In the last decade, CubeSats have matured from educational tools into accepted scientific and commercial assets, with more than 700 of those standardized satellites launched so far. A recent report by the National Academy of Sciences discussed that CubeSats have shown many characteristics of disruptive innovations, like personal computers or cellular phones in the past. However, their current high rates of dead on arrival (DOA) and infant mortality jeopardize this evolution and frequently the underlying cause of this is limited system-level testing done by many developers.

After a definition of the reliability terms and models used, common failures of unmanned past and present spacecraft are described. Although most reliability prediction models use the assumption of random hardware faults, these failure data show that systematic errors, such as failures in design and manufacturing, are the most prevalent source of satellite failure. Also, the increasing utilization of software and the increase in complexity of that software results in more failures in recent missions. The ongoing miniaturization of spacecraft and the professionalization of terrestrial electronics induced the increased use of commercial off-the-shelf (COTS) components for space missions. The qualification of these parts for automotive or industrial purposes make them in many cases also sufficient for space usage, except for vacuum and radiation.

Analyses of current parametric reliability models exposed inconsistencies in the time-dependent behavior of infant mortality and wear-out modelled. New parametric models implemented suggest that the pooled group of satellites of different sizes experiences no distinct wear-out, as their failure rate function has the shape of a right-open bathtub-curve. Splitting the group up into different mass-classes reveals that wear-out is more prevalent in larger satellites, and that the reliability of smaller satellites is dominated by DOA and failures throughout their lifetime. No significant difference can be observed between the reliability of the different sizes of satellites within the observation window. For CubeSats, the on-orbit reliability data were collected from various sources and used to build the so-called CubeSat Failure Database (CFDB). The extraction of the time-dependent failure behavior of this class of satellites proves that DOA and infant mortality are the most prevalent contributors for CubeSat failure.

To prevent future CubeSat missions from experiencing early failure, a reliability assessment method to identify, track, and subsequently solve possible DOA and infant mortality causes was verified on the CubeSat MOVE-II of the Technical University of Munich. The method is based on an adapted reliability growth model and an online, semi-automated Failure Reporting and Corrective Action System (FRACAS). Out of several growth models tested with the failure data of the satellite, the basic exponential and the delayed exponential growth models show the most promising results. Using the growth models, the remaining failures in the system, the space segment, and the remaining critical failures, as well as the on-orbit reliability are estimated. Besides this, methods implemented to maximize the number of beta testers interacting with the satellite in a Test Like You Fly (TLYF) configuration, and approaches how to shift risk upfront were developed. Finally, for future missions, a reliability prediction method to efficiently trade-off design options in early phases is shown.

This page intentionally left blank.

Zusammenfassung

Diese Dissertation beschreibt verschiedene Methoden für die Nachverfolgung, Bewertung und Voraussage der Zuverlässigkeit von CubeSats. Diese Methoden umfassen Tests auf Systemebene und Modellierung des Zuverlässigkeitswachstums über der Zeit. Innerhalb des letzten Jahrzehnts entwickelten sich CubeSats von Ausbildungssatelliten hin zu akzeptierten Plattformen für wissenschaftliche und kommerzielle Anwendungen, und über 700 dieser Satelliten wurden bereits bis zum heutigen Tag gestartet. In einem Bericht der US-amerikanischen National Academy of Sciences wurde CubeSats mehrere Eigenheiten einer disruptiven Innovation, ähnlich zu Computern und Mobiltelefonen in der Vergangenheit, zugesprochen. Die derzeitigen hohen Raten an frühen Ausfällen und an Verlusten ohne Kontakt zum Satelliten, in vielen Fällen begründbar durch mangelnde Tests auf Systemebene, gefährden jedoch diese Entwicklung.

Nach einer anfänglichen Definition der verwendeten Zuverlässigkeitsbegriffe und -modelle werden zunächst Fehler und Fehlerursachen auf Satellitenmissionen erläutert. Beinahe alle derzeitigen Modelle zur Vorhersage von Zuverlässigkeit stützen sich auf die Annahme von zufälligen Bauteilfehlern als Hauptursache des Versagens. Daten aus vergangenen Missionen und Tests am Boden zeigen jedoch, dass systematische Fehler, beispielsweise in der Konstruktion oder in der Fertigung des Satelliten, die häufigsten Gründe von Versagen sind. Auch spielt für die Zuverlässigkeit von Satelliten die vermehrte und intensivere Nutzung von Software, und die gesteigerte Komplexität derselben, eine immer größere Rolle. Die Miniaturisierung unbemannter Raumfahrzeuge und die Professionalisierung terrestrischer Elektronik führen zu einer immer vermehrten Nutzung kommerzieller Komponenten in der Raumfahrt. Die Qualifikation eines Großteils dieser Bauteile für den Automobilbereich oder für breite Industrieanwendungen ermöglicht hierbei die oft direkte Nutzung unter Weltraumbedingungen, hochenergetische Strahlung und Vakuum ausgenommen.

Derzeitige parametrische Zuverlässigkeitsmodelle zeigen bei genauerer Analyse Inkonsistenzen bezüglich früher Ausfälle und Abnutzung. Die für diese Arbeit neu entstandenen, parametrischen Modelle deuten darauf hin, dass die gemeinsame Gruppe von Satelliten verschiedenster Größe keine Anzeichen von Abnutzung in späten Missionsphase zeigt, also die Badewannenkurve nach rechts offen ist. Gleichzeitig werden durch die Aufteilung von Satelliten in verschiedene Massenkategorien Abnutzung als eine Fehlerkategorie in großen Satelliten und eine erhöhte Chance auf Verlust des Satelliten ohne Kontakt für kleinere Satelliten nachgewiesen. Insgesamt existieren innerhalb des gewählten Beobachtungszeitraums keine signifikanten Zuverlässigkeitsunterschiede zwischen den verschiedenen Massenkategorien. CubeSats stellen hiervon eine Ausnahme dar, und um ihre Zuverlässigkeit beurteilen zu können ist die Sammlung von Fehler und Fehlerursachen vergangener CubeSat-Missionen nötig. Die sogenannte CubeSat Fehlerdatenbank (CFDB) wird mit diesen Daten gespeist und dieser Arbeit vorgestellt. Aus der CFDB wird die zeitabhängige Zuverlässigkeit von CubeSats ermittelt. Diese Daten zeigen, dass Frühausfälle sowie Verlust des Satelliten ohne Kontakt die dominantesten Fehlerarten darstellen.

Um Frühausfälle künftiger CubeSat-Missionen zu vermeiden wird eine Zuverlässigkeitsbewertungsmethode zur Identifikation, Nachverfolgung und Lösung von Frühfehlern präsentiert und die Ergebnisse der Verifikation dieser Methode am MOVE-II CubeSat der Technischen Universität berichtet. Die Methode basiert auf einem Zuverlässigkeitswachstumsmodell und einem halbautomatisierten, online verfügbaren

System zur Fehlermeldung und Fehlerkorrekturnachverfolgung. Mehrere Zuverlässigkeitswachstumsmodellen werden mit Testdaten des Satelliten gespeist. Das Basis-Exponentialmodell und das Exponentialmodell mit variablem Startdatum zeigen hierbei die robustesten Ergebnisse. Die Verwendung dieser Zuverlässigkeitswachstumsmodelle erlaubt die Prognose der Anzahl an verbleibenden Fehlern im Gesamtsystem, im Raumsegment sowie die Vorhersage verbleibender kritische Fehler. Auch eine Abschätzung der Zuverlässigkeit kann anhand der gesammelten Testdaten erfolgen. Neben diesem Ansatz werde auch Methoden zur Maximierung der Anzahl an Satellitentestern und zur Umsetzung des „Teste wie du fliegst“ Prinzips (engl. Test Like You Fly) sowie Ansätze zur Risikoverschiebung in frühe Projektphasen gezeigt. Eine Zuverlässigkeitsvorhersagemethode zur effizienten Abwägung verschiedener Konstruktionsvarianten künftiger CubeSat-Missionen bildet den Abschluss der Arbeit.

Contents

Preface	III
Danksagungen.....	V
Abstract	VII
Zusammenfassung	IX
Contents	XI
Abbreviations & Acronyms	XIII
Symbols.....	XVII
1 Thesis Scope.....	1
1.1 Statement of Work	1
1.2 Motivation	1
1.3 Working Hypotheses and Problem Statement	4
1.4 Approach.....	5
2 Reliability of Satellites	7
2.1 Reliability of Satellites	7
2.1.1 Definition of Terms used in Reliability Engineering	9
2.1.2 Classification of Space Missions Failures and their Root Causes	16
2.1.3 Reliability Analysis of Satellites	24
2.2 Reliability Prediction, Assessment and Assurance in Space Missions	40
2.2.1 Reliability Prediction in Space Missions.....	41
2.2.2 Reliability Assessment in Space Missions	47
2.2.3 Reliability Assurance in Space Missions	57
2.3 CubeSats and the ongoing Miniaturization of Satellites.....	66
2.3.1 The ongoing Miniaturization of Satellites and its Implications	66
2.3.2 Risks and Reliability of CubeSat Missions	73
3 Gap Analysis & Objectives	80
3.1 Objectives & Anti-Objectives	82
4 Work and Results	84
4.1 Analysis of Satellite Reliability.....	84
4.1.1 Analysis of Small Satellite Reliability	104
4.1.2 Analysis of Medium Satellite Reliability	112
4.1.3 Analysis of Large Satellite Reliability	118
4.2 CubeSat Reliability	123
4.3 Reliability Assessment and Reliability Prediction of MOVE-II	136

4.3.1	The Development of MOVE-II	136
4.3.2	Assessing the Reliability of MOVE-II	159
4.3.3	Predicting the Reliability of MOVE-II	186
5	Discussion	191
5.1	Analysis of Satellite Reliability	193
5.2	CubeSat Reliability	196
5.3	Reliability Assessment and Prediction of MOVE-II	200
5.4	Recommendations for Building a CubeSat within a University	205
6	Conclusion.....	211
6.1	Summary	211
6.2	Conclusion	214
6.3	Future Work.....	216
7	References	219
	List of Publications	234
	List of Supervised Theses	237
Appendix A	List of Figures and Tables	241
A.1	List of Figures.....	241
A.2	List of Tables	245
Appendix B	Supplementary Figures and Tables.....	246

Abbreviations & Acronyms

2SMARD	Redundant Shape Memory Alloy Hold-Down & Release Mechanism	COTS	Commercial Off-the-Shelf
ADCS	Attitude Determination and Control System	CW	Continuous Wave
ADM	Antenna Deployment Mechanism	DMSP	Defense Meteorological Satellite Program
AF	Acceleration Factor	DOA	Dead On Arrival/Activation
AFRL	US Air Force Research Laboratory	DoD	Department of Defense
AGREE	Advisory Group of Reliability of Electronic Equipment	EBITDA	Earnings before Interest, Taxes, Depreciation and Amortization
ALT	Accelerated Life Testing	ECSS	European Cooperation for Space Standardization
AMSAA	Army Materiel System Analysis Activity	EEE	Electrical, Electronic and Electromechanical
AMSAT	Radio Amateur Satellite Corporation	EKF	Extended Kalman Filter
AOCS	Attitude and Orbit Control System	EM	Engineering Model
APL	Applied Physics Laboratory	EMI	Electromagnetic Interference
AR4JA	Accumulate, Repeat-by-4, and Jagged Accumulate	EO	Earth Observing
ASIL	Automotive Safety Integrity Level	EPS	Electrical Power System
AT	Accelerated Tests	ESA	European Space Agency
BBBW	Beaglebone Black Wireless	ESD	Electrostatic Discharge
BEXUS	Balloon Experiments for University Students	FDM	Fused Deposition Modeling
BOL	Beginning of Life	FEF	Fix Effectiveness Factor
CCSDS	Consultative Committee for Space Data Systems	FIT	Failure per billion hours
CDR	Critical Design Review	FM	Flight Model
CDS	CubeSat Design Specification	FMEA	Failure Mode and Effects Analysis
CFDB	CubeSat Failure Database	FMECA	Failure Modes, Effects and Criticality Analysis
COM	Communication System	FPGA	Field Programmable Gate Array
COPUOS	United Nations Committee on the Peaceful Uses of Outer Space	FRACAS	Failure Reporting, Analysis and Corrective Action Systems

FTA	Fault Tree Analysis	LEOP	Launch and Early Orbit phase
GAO	US Government Accountability Office	LRT	Institute of Astronautics
GEO	Geostationary Orbit	MarCO	Mars Cube One
GEOS	Geodetic Earth Orbiting Satellite	MATED	Model And Test Effectiveness Database
GEVS	General Environmental Verification Standard	MCMC	Markov Chain Monte Carlo
GOCE	Gravity field and steady-state Ocean Circulation Explorer	MCU	Multiple Cell Upset
GOES	Geostationary Operational Environmental Satellite	MEO	Medium Earth Orbit
GPIO	General Purpose Input/Output	MLE	Maximum Likelihood Estimation
GPS	Global Positioning System	MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
GS	Ground Station	MOVE	Munich Orbital Verification Experiment
GSD	Ground Sampling Distance	MRAM	Magnetic Read Only Memory
GSE	Ground Support Equipment	MTBF	Mean Time Between Failure
GSFC	Goddard Space Flight Center	MTTF	Mean Time to Failure
HALT	Highly Accelerated Life Testing	NAS	US National Academy of Sciences
HASS	Highly Accelerated Stress Screening	NASA	National Aeronautics and Space Administration
HDRM	Hold Down and Release Mechanism	NEAR	Near Earth Asteroid Rendezvous
HiL	Hardware-in-the-Loop	NHPP	Non-Homogeneous Poisson Process
I2C	Inter-Integrated Circuit	NOAA	US National Oceanic and Atmospheric Administration
IABG	Industrieanlagen-Betriebsgesellschaft mbH	NRC	US National Research Council
IADC	Inter-Agency Space Debris Coordination Committee	NSF	US National Science Foundation
IEC	International Electrotechnical Commission	NTDS	Naval Tactical Data System
IMP	Interplanetary Monitoring Platform	OBC	On-Board Computer
JAXA	Japan Aerospace Exploration Agency	OPS	Mission Operations Interface
JPL	Jet Propulsion Laboratory	ORS	Operationally Responsive Space
KISS	Keep it simple, stupid	PBL	Project-Based Learning
LDPC	Low-Density Parity-Check	PC	Personal Computer
LED	Light-Emitting Diode	PCB	Printed Circuit Board
LEO	Low Earth Orbit	PDF	Probability Density Function

PDR	Preliminary Design Review	THM	Thermal System
PEM	Plastic Encapsulated Microelectronics	TID	Total Ionizing Dose
PFM	Proto-Flight Model	TLYF	Test Like You Fly
PL	Payload	TRL	Technology Readiness Level
PNZ	Percent-Non-Zero	TRW	Thompson Ramo Wooldridge
PRA	Probabilistic Risk Assessment	TTC	Telemetry, Tracking and Command
QR Code	Quick Response Code	TUM	Technical University of Munich
RADC	Rome Air Development Center	TV	Thermal-Vacuum
RAND	Research And Development	TVAC	Thermal-Vacuum Chamber
RAX	Radio Aurora Explorer	TWT	Travelling Wave Tube
RBD	Reliability Block Diagrams	TWTA	Travelling Wave Tube Amplifiers
RF	Radio Frequency	UHF	Ultra High Frequency
RPN	Risk Priority Number	UNEX	University-Class Explorer
SAD	Solar-Array Deployment	VHF	Very High Frequency
SCU	Space Computer Unit	WARR	Scientific Workgroup for Rocketry and Spaceflight
SEB	Single Event Burnout		
SEE	Single Event Effect		
SEFI	Single Event Functional Interrupt		
SEGR	Single Event Gate Rupture		
SEL	Single Event Latch-up		
SET	Single Event Transient		
SEU	Single Event Upsets		
SHE	Single Event Hard Error		
SLOC	Source Line of Codes		
SPI	Serial Peripheral Interface		
SPN	Stochastic Petri Net		
SSED	Space Systems Engineering Database		
SSTL	Surrey Satellite Technology LTD		
STEREO	Solar Terrestrial Relations Observatory		
STR	Structure & Deployables		
TC	Thermal Cycling		
TDRSS	Tracking and Data Relay Satellites System		

This page intentionally left blank.

Symbols

α	Mixing weight	b	Constant error detection rate per undetected error
α_f	Failure mode ratio	C_m	Criticality number for one failure mode
β	Shape parameter	d	Not constant error detection rate per undetected error
β_{cp}	Conditional probability for loss of function or mission	E_a	Activation energy
β_g	Reliability growth parameter	f	Failure probability density
γ	Location parameter	F	Probability of failure
γ_0	Pre-exponential factor for Arrhenius model	H	Number of estimated errors
θ	Scale parameter	k	Number of different part categories
μ	Mean	K	Quantity of generic parts
λ	Failure rate	m	Mass
\bar{v}	Number of non-repairable items still working	n	Number of independent items
π	Modification factor for part stress methods	p_{NZ}	Ratio of non-zero failure items
σ	Variance	R	Reliability
a	Initial error content	R_r	Temperature dependent reaction rate
a_{af}	Acceleration factor	R^2	Goodness-of-fit
A	Temperature factor for Arrhenius model	t	Time
c	Inflection parameter	T	Temperature

This page intentionally left blank.

1 Thesis Scope

“He Who Can Handle the Quickest Rate of Change Survives.”

– John Richard Boyd: New Conception for Air-To-Air Combat

“We tend to overestimate the effect of a technology in the short run and underestimate the effect in the long run.”

– Roy Amara

1.1 Statement of Work

This thesis originated in the author’s role as project manager for MOVE-II, the Munich Orbit Verification Experiment II, which is the second CubeSat of the Institute of Astronautics (LRT) of the Technical University of Munich (TUM). As many other CubeSats worldwide, MOVE-II is primarily an educational project – designed, built, and (to be) operated mainly by students. The 10 x 10 x 13 cm¹, 1.2-kilogram satellite will be launched into space in October 2018, and the main mission should last 6 months. To ensure successful operations and prevent the mission from dead on arrival (DOA) or early failure, several methods to shift risk upfront were applied during the development, and the reliability of the satellite was assessed during system level testing. Thus, the main scope of this work is to describe the applied methods and the assessment and present lesson learned of that process. It is the hope of the author that these methods and the developed assessment approaches can also help other CubeSat developers, not just at universities, and will lead to a reduction of DOA and infant mortality cases of CubeSats.

1.2 Motivation

The idea behind CubeSats goes back to 1999, when Bob Twiggs of Stanford University and Jordi Puig-Suari of California Polytechnic University invented this standardized miniature satellite class, primarily for educational purposes [1]. Over the past 19 years, CubeSats evolved from educational tools to accepted platforms for scientific and commercial applications. In the current decade, this trend has accelerated, and, according to a 2016 report from the Space Studies Board of the US National Academies of Sciences (NAS), over 80% of all science focused CubeSats were launched between 2010 and 2016. Also, more than 80% of peer-reviewed papers on science on CubeSats originated from post 2010 [2]. The increasing involvement

¹ For CubeSats, this envelope is standardized and called 1 Unit (1U).

of private companies in spaceflight, broadly summarized as NewSpace [3], further boosted the number of launched and planned CubeSats and small satellites². According to Euroconsult [4], 60% of the 220 satellites launched in 2016 were below 500 kilograms, and among these, 50% were CubeSats³. Recently, the US National Oceanic and Atmospheric Administration (NOAA) investigated the usage of weather data provided by CubeSats [6], which could cut costs for weather forecasting by 95% [7]. Planet, a private US company, has launched more than 250 CubeSats for optical imagery so far [8] and in May 2018 the Jet Propulsion Laboratory (JPL) successfully launched two CubeSats, called Mars Cube One (MarCO), on a piggyback mission to Mars to establish a real-time Mars relay to the InSight lander during Mars reentry [9].

These examples show us the broad applications and acceptance of CubeSats, but they also tell us something about the future role of CubeSats and small satellites. Space business itself is growing, potentially reaching a market value of up to US\$30.1 billion in the next decade (from US\$8.9 billion in the previous decade) [4], and in many applications, CubeSats can be seen as an addition and not as a replacement to most of the traditional, big missions. Or, as NOAA puts it, “*Small satellites are another tool in the tool chest that we might consider using to meet our operational requirements*”⁴.

The miniaturization and increased utilization of commercial off-the-shelf (COTS) parts led to growth trend, more or less equivalent to Moore’s Law, of ground sampling distance (GSD), data rate, and data volume of small satellites between 1990 and 2010 [10]. However, there are also problems arising in the small satellite and CubeSat domain. In 2013, Swartwout [11] reported on-orbit failure rates of more than 50% among university-led CubeSat missions. In his study 112 CubeSats, launched until the end of 2012 were analyzed and 19 of them suffered from launch failures, and out of 93 successfully placed on-orbit, 48 failed. Many of the 48 failures were either DOA or failing early into their missions. His study was one of the starting points of this thesis.

The limited lifetime of CubeSats, due to an accepted higher risk and the approach that usage of commercial off-the-shelf electronics “as-is”, is acceptable, and, as we will see later, somehow even desirable due to accelerating innovation cycles. The current rates of DOA and early failures described by Swartwout, however, are not acceptable, especially for commercial and scientific missions and at their current rate not even for university missions, in which usually some mission operations with the satellites must be achieved to obtain some of the mission goals. Moreover, a study by the author of this thesis on CubeSat reliability [12] showed a mismatch between self-perception and perception of other university-built CubeSats. A group of 88 CubeSat developers estimated the likelihood of failure for their own CubeSat in the projected lifetime to be slightly above 16% on average. Asked about the likelihood of failure for a general university-built CubeSat within the first 6 months, the same group estimated the chance to be slightly below 50% on average. That is a difference of 34% between self-perception and perception of other developers. These results and the study will be covered more extensively in Section 4.2.

Traditional methods for assuring reliable space systems are not possible when dealing with limited resources, and often limited experience, in university teams. Reliability, in traditional missions, is achieved by the reliance on high-reliability, space-proven parts, redundancy, a strict test program, and corrective actions from ground as soon as the spacecraft is launched [13]. The limited envelope of CubeSats and limited resources/time prevent many university teams from applying those traditional methods. Thus, as CubeSats are placed at one extreme of satellite design, production, and testing, university-built CubeSats will be at the center of attention in this thesis. Interestingly, CubeSats can be also seen as a reincarnation of the Faster-Better-Cheaper program [14] of the National Aeronautics and Space Administration (NASA),

² In general, small satellites are broadly defined as satellites below 500 kg. We will discuss this in Section 2.3.

³ According to Swartwout [5], 77 CubeSats were launched in 2016.

⁴D. Werner, NOAA sees great promise and challenges in using data from small satellite constellations - SpaceNews.com. [Online] Available: <http://spacenews.com/noaa-smallsat/>. Accessed on: Jan. 09, 2018.

which was conducted between 1992 and 2000. Many lessons learned regarding limited reliability during this program can also be applied to CubeSats, as discussed further in Section 2.3.

Statistically, the space industry always had the dilemma of a limited, often single digit number of satellites being produced, making conventional statistical analysis, as done in terrestrial applications, not possible. The precision and statistical significance of data was always restricted by the small sample size from similarly produced satellites [15]. NewSpace, and the arising economical aspects of spaceflight, will lead to new questions in terms of production and satellite reliability itself, and planned higher production numbers might change established approaches in the future. Historically, economic loss was a consequence of the lack of reliability of satellites [13]. In NewSpace, there is also a close relationship between economic loss and innovation cycles, in which loss can also have its origin from a too late adaption of new technology. Traditionally, satellites were almost always highly customized, hand-made products. In NewSpace, so-called mega-constellations such as OneWeb try to produce and use satellites in a four-digit scale while trying to cut satellite production costs by 90% versus today's costs [16]. Furthermore, NewSpace also comes with the promise to bring down launch costs for small satellites – either by reusable rockets, or by dedicated small satellite launcher.

CubeSats could be significant idea generators for this cost cutting efforts in satellite production and operations by using COTS state-of-the-art electronic parts and exploring new methods of automated production, testing and mission operations. Thereby, the reliability increase of automotive and industrial COTS parts over the last decades is one main pillar of the success story of miniaturized spacecraft. According to the aforementioned NAS report, CubeSats have shown many characteristics of disruptive innovations, similar to personal computers or cellular phones [2]. CubeSats can be utilized to bridge the so-called Technological Readiness Level (TRL) “Valley of Death”. This valley, caused by the significant resources needed to bring technology from TRL 6 to TRL 7⁵, is one reason for the heavy reliance on heritage technology in traditional space missions. Furthermore, advancing non-validated technology to a flight-ready state is a primary cause for budget overruns and time delays in many NASA and Department of Defense (DoD) missions. A US Government Accountability Office (GAO) report shows the significant cost growth (near 35%) which 52 programs experienced on average using immature technology. Bridging this valley is possible by using CubeSats, as shown by the MCubed-2 mission, which flew a new radiation-hardened-by-design Field Programmable Gate Array (FPGA) for the first time in 2013 [17], [18].

Thus, the evolution of CubeSats into a disruptive innovation and the advent of NewSpace are additional underlying motivation factors for this thesis. The high rate of early failure and DOA in university-built CubeSats must be overcome, if CubeSats shall become professional assets of the NewSpace era. Yet, the limited resources and required fast delivery of those small satellites limit the application of traditional methods in educational university satellite projects. Designing and testing spacecraft usually involves difficult judgement calls [19], and those calls are harder to make with the limited envelope and resources of a CubeSat mission. Additionally, many of the university teams lack the necessary experience to make those calls on a sound basis and currently, there is no agreed standard on CubeSat mission assurance [20]. CubeSats often do not have the possibility to switch to redundant systems, and often cannot perform their mission in a degraded state, which is different from traditional satellites [21]. Finally, many of the failed CubeSat missions not only led to unfulfilled scientific, tech-demo or commercial goals, but also produced a not negligible amount of space debris – a topic discussed in more detail in Chapter 5.

⁵ According to NASA, TRL 6 means a system/subsystem model or prototype demonstration in a relevant environment (ground or space), while TRL 7 describes a system prototype demonstration in space environment [17].

1.3 Working Hypotheses and Problem Statement

The motivation presented in the last section leads us to three consecutive hypotheses for this work. First, the data of past satellite missions have to be analyzed regarding their different time-dependent failure behavior. Hence, the first working hypothesis is:

The time-dependent failure behavior of satellites, namely dead on arrival, infant mortality, random failure and wear-out, can be individually extracted from today's in-flight reliability data.

Having analyzed and extracted the time-dependent failure behavior, the next step is to study the differences between CubeSats and traditional commercial satellites. Thus, the second working hypothesis is:

The failure behavior is substantially different between commercial satellites and CubeSats, and can be quantified.

Overall, the goal of this thesis is to reduce the chance of infant mortality and DOA for future university-built CubeSat missions. Hence, the third working hypothesis is:

A test strategy for CubeSats can be developed to identify and solve possible DOA and infant mortality causes and thus significantly and efficiently increase their reliability.

There is abundant statistical data on satellite missions since the beginning of spaceflight. These data show that, once a satellite is deployed and successfully tested out in space, it will have a chance of over 90% to achieve its projected lifetime [19]. Research carried out by Saleh and Castet [22] on 1,584 spacecraft, launched between January 1990 and October 2008, is one of the richest sources of time-dependent failure behavior of traditional satellites. We will analyze their data in Subsection 2.1.3, and come back to it in Section 4.1, in which we try to evaluate the first working hypothesis.

Although a lot of effort is put into the reliability of parts used for traditional missions, data from past missions also show that most spacecraft fail due to common cause, or systematic cause, relating to design and engineering errors rather than random hardware faults. Analysis by the Aerospace Company shows that 73% of all failures are due to systematic faults relating to design errors [23]. This has serious implications on the methods how we predict spacecraft reliability and on the fault management systems that should ensure it, since traditionally, reliability prediction methods such as the MIL-HDBK-217F [24, 25] strongly rely on random part failure and the absence of failures in design and workmanship, and so do most fault management systems. Of course, the absence of part failures in most missions could also mean that the effort spent on part assurance is mitigating most part-related errors and more efforts should be put in place to ensure the same for systematic failures.

The miniaturization of satellites and the increased utilization of COTS parts lead to other open questions. Automotive electronics, heavily used in CubeSats and small satellites, nowadays undergo a rigorous screening process and are typically tested for harsh environments, similar or more extreme than the environment in low earth orbit (LEO)⁶ [26]. Additionally, other than space rated parts, automotive and industrial parts are produced in statistically relevant quantities for reliability estimation and nowadays often have similar failure rates as space parts. With the miniaturization of satellites, the number of parts decreases. The lower amount of parts decreases the complexity and should thereby decrease the failure rate. Fleeter [27] used a simple exponential model for that assumption, stating that smaller satellites could

⁶ Except for vacuum and high energy radiation – we will address both topics in Subsection 2.3.1.

use less reliable parts (quantified by reliability R_0) and achieve the same reliability R as larger, more complicated satellites due to their reduced number of overall parts n :

$$R = R_0^n \quad (1)$$

According to Fleeter a less complex spacecraft should also reduce the human error rate in design and manufacturing [27]. Sarsfield [28] noted that the spaceflight community is divided by the question if either a single string, simple or a larger, more redundant spacecraft will be more reliable. He reported that overall, the combination of high reliability COTS parts with simpler and smaller spacecraft leads to reduced electrical loads and launch loads. Thus, the smaller loads should increase the reliability of small satellites. Data about the success and failure rate of CubeSats [11], already mentioned in the last section, currently clearly show that the opposite is the case for CubeSats. Although Swartwout analyzed the success and failure rates, and rightly presented the problem of infant mortality and DOA, no time-dependent failure behavior of CubeSats is provided by his work.

Therefore, the time dependent failure-behavior of CubeSats must be identified along with the underlying reasons causing this behavior. In traditional missions, established satellite producers can rely on an abundance of lessons learned for their projects [29]. University-based CubeSat developers, as the MOVE-II team, lack this resource as well as general experience. Therefore, they need methods to assess the reliability of their satellite while testing it to make sound project management decisions. These decisions also include which system level tests are to be preferred over others, considering the limited resources in such projects.

The effort of NASA in the 1990s to build faster, better and cheaper spacecraft was already mentioned. Nowadays jokes about the program such as “Faster-Better-Cheaper – pick any two” exist. Besides technical and management challenges, also a cultural challenge is currently occurring in spaceflight. The “failure is not an option” belief of Apollo is still reinforced in many spaceflight programs. A cheap product for space use is generally expected to fail, a prejudice that can also be observed for terrestrial products, and an expensive product is expected to last longer [30]. Similarly, some traditional satellite developers consider CubeSats as “space debris with antennas”. If we do not reduce the high infant mortality and DOA rate in the future, while continuing being a disruptive technology, we are going to prove them right.

1.4 Approach

To reduce the infant mortality and DOA rates of CubeSats, a broader range of topics has to be explored and then narrowed down subsequently. The most promising approaches were evaluated on the MOVE-II CubeSat, also with the goal to reduce the satellites’ own chance for early failure. As presented earlier, university-built CubeSats have to deal with some unique characteristics of the educational environment during development and often lack resources and experience of the involved persons. Nevertheless, some approaches and conclusions might also be useful for larger satellites.

Chapter 2 will introduce fundamentals on reliability engineering and explain which root causes exist for failures of spacecraft. To this end, it is also important to look at historical failure rates and time-dependent failure behavior of satellites, and we will try to observe patterns in these data. Fundamentals on reliability prediction, reliability assessment and reliability assurance will also be covered in Chapter 2, since it is important to understand the differences between them and the range of applications of all three. Examples from past space missions will support that chapter. In the last section of Chapter 2 we will focus on the ongoing miniaturization of spacecraft and the strongly related increased use of COTS parts in those small missions. Both have implications on the reliability and risk of those spacecraft, and a review of historical data will show the current gaps within that field.

These identified knowledge gaps are then presented in Chapter 3. As aforementioned, CubeSats are currently suffering from an excessive rate of DOA and infant mortality cases, and many of those cases can be attributed to poor system level testing. To reduce this rate, the biggest gap to be closed by this work is to improve the development and reliability assessment process of CubeSats. Time-dependent failure behavior of past CubeSat missions had to be collected for that purpose, and parametric fitting of larger missions had to be studied. Minor gaps exist in the lack of Failure Reporting and Corrective Action Systems (FRACAS) for CubeSats and in reliability prediction methods used for most current CubeSat missions.

We will then tackle the problem from large to small and begin with the analysis and extraction of the time-dependent failure behavior of satellites from today's in-flight reliability data. The major source of data for this analysis will be a group of 1,584 satellites studied by Saleh & Castet [22] and other authors. Having learned about the behavior of larger satellites, we will continue our research with an analysis of past CubeSat missions, and present our CubeSat Failure Database (CFDB) that was built for that purpose. Parametric and nonparametric descriptions of the on-orbit reliability of the studied group of 178 CubeSats will help to determine the current patterns of CubeSat failure. To increase our chance of identifying, tracking, and resolving bugs, FRACAS and methods to improve the Test like You Fly (TLYF) approach were studied and applied on MOVE-II. This was combined with selected reliability assessment methods to estimate how many potential failures were left in the system at a specific point in time. Looking at future applications, reliability prediction methods suitable for design-tradeoffs in CubeSat missions are finally presented.

In the discussion, the applicability to other (also larger) missions but also the need for more data is specified. After concluding the work, several open topics for future research are presented as this work is seen only as a first step to reduce the infant mortality and DOA rates of future CubeSat missions.

2 Reliability of Satellites

“Tempus edax rerum” – “Time, the devourer of all things”

– Ovid

“The major difference between a thing that might go wrong and a thing that cannot possibly go wrong is that when a thing that cannot possibly go wrong goes wrong it usually turns out to be impossible to get at or repair.”

– Douglas Adams: The Hitchhiker’s Guide to the Galaxy

In this chapter we will start with basic definitions used in reliability engineering and classifications of spacecraft failures. We will then look back in time and analyze the historical failure rates of unmanned space missions and try to find patterns in the collected data. A section on the fundamentals of reliability prediction, reliability assessment, and reliability assurance is followed by background on CubeSats and the ongoing miniaturization of spacecraft, focusing also on risks of CubeSat projects and the reliability of CubeSats.

2.1 Reliability of Satellites

Reliability and spaceflight are tightly connected since the beginning of the space age. As presented in the first chapter, spacecraft often present a unique set of properties, being in a very small, single digit sample size, or, in many of the cases, a one-of-a-kind single unit. Spacecraft are designed to work in a hostile environment that cannot be fully recreated here on earth for testing purposes. Thus, they can also be seen as one-shot items. Testing is usually done on the spacecraft itself, the so-called flight model (FM), or, where time and resources allow, on an engineering and/or qualification model (EM or EQM) beforehand. On the FM, failures and malfunctions are corrected during testing, if possible, and the spacecraft is launched afterwards. Conventional statistics, dealing with large sample sizes and assigning probabilities of success and failure to certain items, are thus not feasible for satellites [31], [32]. The satellite industry, by the time of this thesis, lacks “satellite mass production”, in which four or five digit numbers of identical satellites are produced for the same operational environment – and their life data subsequently statistically analyzed [15]. This situation could change in the future with the advent of mega-constellations. Generally, new products appear on the market in a never-seen before pace in our modern society. Reliability of modern terrestrial applications is mandatory, and manufacturers have to decide on certain trade-offs between cost and product reliability to satisfy customers. As the complexity increases, product life cycles are getting shorter with each generation. Although achieving reliability in these short cycles is costly and difficult, putting an unreliable product on the market can have far worse consequences for a manufacturer in our globalized and connected world [33]. Ultimately, all products will degrade and fail someday. In terrestrial applications, failures and degradation can be prevented by maintenance or replacement. In space, this can be done only on a limited scale. Although the Space-Shuttle maintenance missions to the Hubble Space Telescope and

the replacement of satellites of the Global Positioning System (GPS) with on-orbit spares are examples for that, both are currently not very common ways to deal with reliability issues in spaceflight. Again, this could change in the future with mega-constellations⁷ and planned robotic on-orbit servicing missions.

Similar to terrestrial applications, customers expect a certain reliability of their spacecraft. There is also a broad spectrum of possible options for customers, ranging from short-term CubeSat missions (lifetime of months to years) to geostationary platforms (lifetime of 15-20 years). Other than in terrestrial applications, traditional spacecraft production prioritizes reliability over cost and schedule in many cases, and in some of them the “failure is not an option” approach led to extreme resource overrun and delays. In the view of the author, “failure is not an option” is the right way for manned missions and projects with demanding scientific goals, but it cannot be the only choice for unmanned spaceflight. We will revisit this thought when dealing with the Faster-Better-Cheaper approach in Section 2.3.

There are many reasons besides cost and schedule overrun why satellite manufacturers will not prefer to go for the longest possible lifetime. Although there is research done on the feasibility of 100-year lifespan space missions [35], obsolescence of the launched technology, unanticipated failure modes in the hostile space environment, and higher satellite production and launch costs are reasons to not go for the longest possible lifetime. Higher satellite production and launch costs are caused by the additional redundancy, fuel, batteries and solar cells needed for longer lifetime. Also, in some cases, a higher mass on the bus-side of the satellite leads to a reduced mass on the payload-side [19]. Obsolescence in spaceflight is thereby not only a relevant topic in case the technology launched will be obsolete at some point in the future. It is also important to consider obsolescence in terms of the parts used, i.e., electronic parts not being produced anymore but necessary for some space missions.

The heavy reliance on heritage in spaceflight, combined with the ever-increasing pace of product lifecycles in terrestrial applications, is a problem space-agencies and satellite manufacturer already have to deal with. As heritage can only be fully applied on part-level, since as few as possible changes should happen to the design or the manufacturing processes⁸ [36] of the electronic part, most of the current solutions imply the purchase of 4- or 5-digit numbers of electronic parts and stock keeping. This stock keeping tends to block innovation. Once components or subsystems are “space-qualified”, it becomes very hard to option for design upgrades or changes in later missions. Thus, processors or other electronic hardware flying on current space missions could be already obsolete for a decade or more in terrestrial applications [19]. State-of-the-art COTS electronics are not only faster and more power-efficient than their predecessors, they also often combine functionality, leading to a reduced number of parts, allow novel failure-correction and hard- and software redundancies. At lower cost and reduced footprint, it might be more feasible to have one state-of-the-art integrated COTS chip with four processors in hot redundancy, than one space-qualified, 15-year old single processor chip. This could also be seen as a way to reduce the increasing number of electronic parts used in space missions and therefore reducing complexity and chance of failure. Figure 2-1 shows the electronic parts used for space missions over time and the trend towards increasing complexity and functionality of today’s missions [37].

⁷ For example OneWeb plans to produce about 250 spares for its mega-constellation of 648 satellites [34].

⁸ To have electronic parts produced in the same lot is especially important for radiation tests.

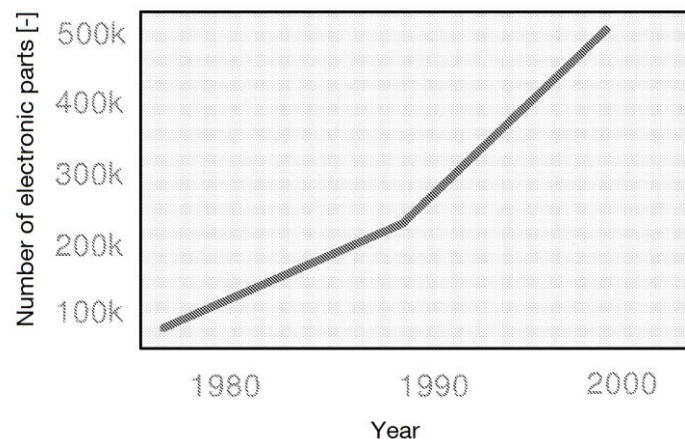


Figure 2-1: Electronic Parts Count in Space Missions over Time. Adapted from: [37].

To summarize, reliability of spacecraft has many unique characteristics to consider. Traditional satellites are being built with increasing complexity, relying heavily on heritage parts kept in store over one or more decades. Reliability is traditionally preferred over schedule and cost and most spacecraft developers desire the highest amount of reliability possible for their system. However, as in terrestrial applications, termination of a program due to cost overrun or delay in the mission can also be seen as reduced system reliability. On each day in which the system is not ready to use when it should be, the reliability of the system is zero. Thus, traditional efforts to achieve high reliability are sometimes counterproductive [38]. CubeSats could be a way to change this, enabling fast missions and space-qualifying novel electronic hardware in short intervals. They could show us ways to design, produce and test spacecraft that fulfill their reliability goals within their limited lifetime, and not more⁹. Before we elaborate on that further, we first have to define later-used terms of reliability.

2.1.1 Definition of Terms used in Reliability Engineering

Reliability, and how to predict and assess it, involves a set of terms to be defined in this subsection. To facilitate reading, new terms are printed in bold in this section. **Reliability** itself can be defined by “*the probability that it [an item] will perform its required function under given conditions for a stated time interval*”¹⁰. Redundant parts may fail during the item’s lifetime, and may be repaired, but the overall system remains functional over the mission duration. Therefore, it is important to distinguish between repairable and non-repairable systems (spacecraft are considered as non-repairable in this thesis), and define the operating conditions, the function and the intended lifetime of the system when stating reliability in a numerical sense (e.g., $R = 0.995$) [39]. Defining reliability as a probability also implies that failures are inevitable, and typically assessed by the so-called mean time to failure (MTTF) [40]. Powell [41] argued that reliability should be considered by engineers as a physical property of the designed system, in accordance with certain physical laws. It can only be designed into a device, similar to volume, mass and other physical properties. The International Electrotechnical Commission (IEC) defines reliability as “*the probability that the product (system) will perform its intended function for a specified time period when operating under normal (or stated) environmental conditions*”¹¹.

⁹ Of course, while being also compliant to the International Space Debris Mitigation Guidelines.

¹⁰ A. Birolini, Reliability Engineering: Theory and Practice, 7th ed. Berlin, Heidelberg, s.l.: Springer Berlin Heidelberg, 2014, pp. 2.

¹¹ D.N.P. Murthy, M. Rausand, and S. Virtanen, “Investment in new product reliability,” Reliability Engineering & System Safety, vol. 94, no. 10, pp. 1593–1600, 2009, pp.1593.

As we have seen in the previous section, it is relatively easy to get statistically significant data on reliability of terrestrial products due to the usually large production batches. For spacecraft, this happens to be a more difficult challenge [28]. In terms of space missions, Hecht [13] defined reliability in spacecraft as mission reliability, meaning the “*probability that at least the essential mission elements will survive*”¹². This considers the fact that many spacecraft rely on redundancy on part and subsystem level. Maurer [36] noted that a stated reliability implies three assumptions: the acceptance of the probabilistic concept of reliability (also admitting the possibility of failure); the concept that system parameters deteriorate slowly with time; and that judgment is necessary to determine the proper state of environmental conditions. Hecht & Hecht [21] further differentiated between the concept of basic reliability in space missions, meaning “without failure of any kind”, and the before described mission reliability, meaning “failure not impairing the mission”. Reliability in space programs is often linked to **risk**. Sarsfield [28] noted that “reliability” will be used by engineers to describe components or systems and “risk” will be used by managers to describe programs. Thus, risk is a higher-order term than reliability. He further described that spacecraft reliability is determined by many factors, has often an inverse proportional relation to complexity, but also that reliability is highly dependent upon how a spacecraft is designed and tested. The successful series of 40 out of 41 Radio Amateur Satellite Corporation (AMSAT) small satellites proves that simplicity and robustness of low-cost designs sometimes outperforms the fragility of more complex and expensive ones in terms of reliability [28].

Unreliability, manifesting itself through failures, is a measure of the lack to perform properly when needed. When a device does not perform as it should, it is said to have failed [42]. The Aerospace Corporation [23] defines **failure** as an unexpected response, in which the function is not recoverable. They further differentiate between failure and **fault**, since the latter one is describing recoverable¹³, unexpected responses [23]. Sometimes the term **anomaly** is used instead of fault, but in order to be consistent in this work, either fault or failure will be used, if possible. Again, in contrast to terrestrial systems, once a failure has occurred in a spacecraft, there is usually no possibility to repair or replace hardware. Failures and faults are sometimes grouped together and then called **malfunctions, problems or errors**. All of them don’t necessarily define whether the function was recoverable or not and are used in different ways in the literature. We will see all terms when dealing with past reliability data. Due to the involved cost and their limited number, space systems are usually not tested to the point of failure to evaluate a new design, in contrast to terrestrial ones. And finally, failures can normally only be analyzed through telemetry sent by the spacecraft and test data gained in ground tests [28].

Historically, failures of spacecraft are economically equal to a complete replacement, including launch (and other costs, such as administrative ones) [13]. Considering the high cost associated with traditional missions, “failure is not an option” seems understandable. On the other hand, the US-company Planet launches CubeSats on a regular basis and uses an evolutionary development approach for that, much as in modern software, in which they can afford to lose satellites and are in fact calculating their business models with that. Thus, the company was able to cope with the loss of 34 satellites in two rocket crashes in 2015 and 2016 [43]. Launch failures are an important, but sometimes forgotten unique characteristic of spacecraft that must not be neglected when discussing satellite reliability – we will do that in Section 2.2. In general, the reliability of more than one item is monitored in many cases and then failures of the group are analyzed over time, using the so-called failure rate. Figure 2-2 shows a representative chart of still operating items out of a group at time t [39].

¹² H. Hecht, “Reliability During Space Mission Concept Exploration,” in Space mission analysis and design, W. J. Larson and J. R. Wertz, Eds., 2nd ed.: Kluwer Academic Publishers, 1998, pp. 704.

¹³ Either by fixing it directly, redundancy or managing around it.

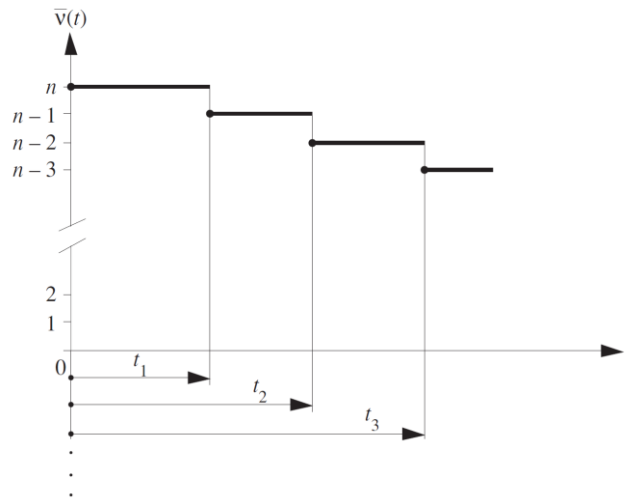


Figure 2-2: Number $\bar{v}(t)$ of non-repairable items still working at time t . Image Source: [39]

The **failure-free time**, depicted as t_1 in Figure 2-2, is the time interval in which no failure occurred in the system(s). Birolini noted that this time is often reasonably long, but it can also be very short due to, for example, failures that happen directly after first power-on. An important assumption is that in general, the item is free of defects and systematic failures at $t = 0$ [39]. We will come back to this later while researching the gathered on-orbit reliability data of satellites.

As we have seen, reliability is a probabilistic concept. Thus, the question of whether an item will operate without failures for a stated period of time can only be answered with probabilities. Again, following the definition of Birolini [39], the true value of reliability can be approximated by:

$$\hat{R} = \frac{\bar{v}(t)}{n} \quad (2)$$

where n is a number of independent items, which are put into operation at time $t = 0$, and $\bar{v} \leq n$ is a number of items accomplishing the desired mission. \bar{v}/n converges with increasing n to the true value of the reliability. Birolini also defined the failure rate $\lambda(t)$, if $R(t)$ is derivable:

$$\lambda(t) = \frac{-dR(t)/dt}{R(t)} \quad (3)$$

Hence, if all items at $t = 0$ are operational (i.e., $R(0) = 1$) [39]:

$$R(t) = \exp\left(-\int_0^t \lambda(x)dx\right) \quad (4)$$

With those definitions, we can also define the probability of failure $F(t)$ as:

$$F(t) = 1 - R(t) \quad (5)$$

and the probability density function (i.e., the relative likelihood that the value of the random variable would equal that sample) $f(t)$ as:

$$f(t) = \frac{dF(t)}{dt} \quad (6)$$

The relationship between these measures of reliability is as depicted in Table 2-1, adapted from a datasheet from the European Power Supply Manufacturers Association [44]:

Table 2-1: Measures of reliability. Adapted from: [44]

	$F(t)$	$R(t)$	$f(t)$	$\lambda(t)$	
$F(t) =$	$F(t)$	$1 - R(t)$	$\int_0^t f(x)dx$	$1 - \exp\left(-\int_0^t \lambda(x)dx\right)$	Probability of failure
$R(t) =$	$1 - F(t)$	$R(t)$	$\int_t^\infty f(x)dx$	$\exp\left(-\int_0^t \lambda(x)dx\right)$	Reliability
$f(t) =$	$\frac{dF(t)}{dt}$	$-\frac{dR(t)}{dt}$	$f(t)$	$\lambda(t)\exp\left(-\int_0^t \lambda(x)dx\right)$	Probability density function
$\lambda(t) =$	$\frac{\frac{dF(t)}{dt}}{1 - F(t)}$	$-\frac{d(\ln R(t))}{dt}$	$\frac{f(t)}{\int_t^\infty f(x)dx}$	$\lambda(t)$	Failure rate

For many applications it is interesting to look at the time-dependent behavior of the failure rate $\lambda(t)$. In the simplest models, a constant failure rate $\lambda(t) = \lambda$ is assumed. In Section 2.2 we will see that this is also assumed in most traditional predictive reliability models, such as MIL-HDBK-217F. If $\lambda(t) = \lambda$, the reliability $R(t)$ is [39]:

$$R(t) = \exp[-(\lambda \cdot t)] \quad (7)$$

However, field data showed that most systems don't experience a constant failure rate. Weibull (and also others before him) noted that many applications experience increasing or decreasing failure rates over time, which can be modelled by the so-called two-parameter Weibull distribution [45]. The failure rate of the two-parameter Weibull distribution is [46]:

$$\lambda(t) = \frac{\beta}{\theta} \cdot \left(\frac{t}{\theta}\right)^{\beta-1} \quad (8)$$

Hence, $R(t)$ is:

$$R(t) = \frac{f(t)}{\lambda(t)} = \frac{\lambda(t) \cdot \exp\left[-\left(\frac{t}{\theta}\right)^\beta\right]}{\lambda(t)} = \exp\left[-\left(\frac{t}{\theta}\right)^\beta\right] \quad (9)$$

The so-called slope or shape parameter β determines which member of the Weibull family of distributions is most appropriate [47]. While varying β , it is possible to model a variety of different distributions, including the exponential distribution for $\beta = 1$. For $0 < \beta < 1$, the failure rate decreases over time, thus early failures,

also called infant mortality, can be captured by this type of Weibull function. $\beta > 1$ means an increasing failure rate, often called wear-out. This flexibility makes the Weibull distribution the most widely used distribution in reliability applications [48]. For wear-out, it can be further distinguished between [22]:

$1 < \beta < 2$	Increasing concave failure rate
$\beta = 2$	Increasing linear failure rate, equivalent to the Raleigh distribution
$\beta > 2$	Increasing convex failure rate
$\beta > 3.5$	Function approaches the normal distribution

The second parameter of the function is the so-called scale parameter θ , which is sometimes also called the characteristic life. Changing θ while β is held constant will alter the function in the direction of the ordinate. Enlarging θ means stretching the failure distribution over a longer time, while reducing θ will compress it in time, thus affecting the probability of failure over time [45]. Statistically, at the time $t = \theta$, 63% of the units will have failed. This is due to:

$$R(t) = \exp\left[-\left(\frac{t}{\theta}\right)^\beta\right] = \exp[-(1)^\beta] \sim 0.37 \text{ (for all } \beta) \quad (10)$$

The effects of different shape and scale parameters are depicted in Figure 2-3, which is adapted from [49]:

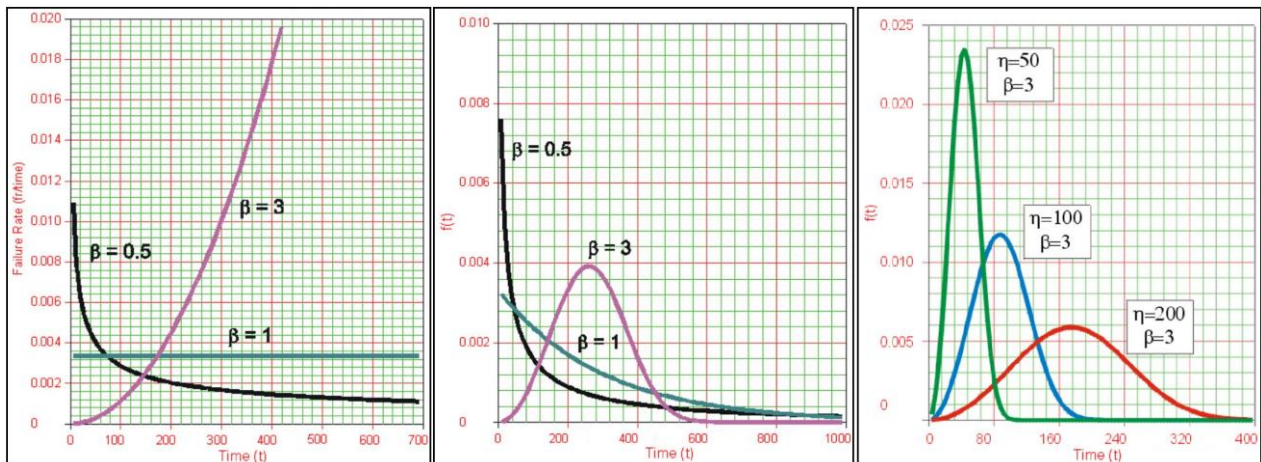


Figure 2-3: Left figure depicts effects of different shape parameters on the failure rate, middle figure shows the probability density function (pdf) for varying shape parameters (while the scale parameter is held to a constant value of $\eta = 300$ days in both cases). Right figure shows effects of different scale parameter (in [49] named η instead of θ) on the pdf, while the shape factor is held constant. Adapted from [49].

The failure rate of large populations of statistically identical items can also be described by a mixture of three different Weibull distributions, widely known as the bathtub curve (see Figure 2-4). The first phase (1.) is dominated by the occurrence of non-deterministic early failures, caused by weakness in materials, components, or the manufacturing process. Deterministic failures, such as design errors, are usually not considered in this phase, since they are already manifesting themselves at $t = 0$ [39]. This will have implications on the time-dependent failure behavior of spacecraft, which we will discuss later. In practice, the early failure period can be between a few hours and 1,000 hours, and is usually eliminated by a burn-in period. The second phase (2.) describes the constant failure rate of the so-called “useful life” of items. Failures in this period are Poisson-distributed. In the third phase (3.), the failure rate increases since more degradation phenomena occur over time. This is called wear-out and can happen sometimes more than 10 years from beginning-of-life for electronic hardware [39].

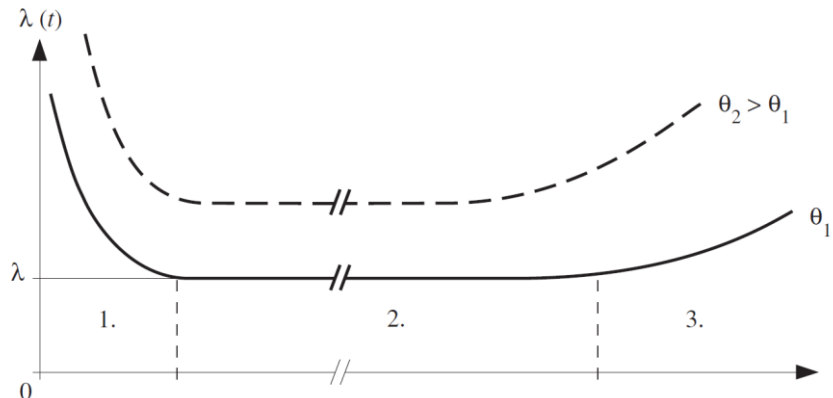


Figure 2-4: The bathtub curve with infant mortality (1.), constant failure rate (2.) and wear-out (3.) The failure rate of the dashed line is higher due to the product being exposed to a more extreme operating environment. Image Source: [39]

Besides the Weibull distribution, there are also other distributions that are commonly used in reliability engineering to describe time dependent failure behavior. Table 2-2, adapted from [36], gives an overview. The mathematical description of the most common distributions is also depicted in Table 6-1, Appendix B.

Table 2-2: Statistical distributions used in reliability engineering. Adapted from [36]

<i>Statistical Distribution</i>	<i>Fields of Application</i>	<i>Examples</i>
Normal	Various physical, mechanical, electrical, chemical properties	Capacity variation of electrical capacitors; tensile strength of aluminum alloy sheet; monthly temperature variation; penetration depth of steel specimens; rivet-head diameters; electrical power consumption in a given area; electrical resistance; gas molecule velocities; wear; noise generator output voltage; wind velocity; hardness; chamber pressure from firing ammunition
Log-normal	Life phenomena; asymmetric situations where occurrences are concentrated at the tail end of the range, where differences in observations are of a large order of magnitude	Automotive mileage accumulation by different customers; amount of electricity used by different customers; downtime of a large number of electrical systems; light intensities of bulbs; concentration of chemical process residues.
Weibull (two-parameter)	Same as log-normal cases. Also, situations where the percent occurrences (say, failure rates) may decrease, increase, or remain constant with increase in the characteristic measured, for parts at debug, wear-out, and chance failure stages of product's life.	Life of electronic tubes, antifriction bearings, transmission gears, and many other mechanical and electrical components; corrosion life; wear-out life

Exponential	The life of systems, assemblies, etc. For components, situations where failures occur by chance alone and do not depend on time in service, frequently applied when the design is completely debugged for production errors	Vacuum-tube failure life; expected cost to detect bad equipment during reliability testing; expected life of indicator tubes used in radar sets; life to failure of light bulbs, dishwashers, water heaters, clothes washers, aircraft pumps, electric generators, automobile transmissions
Binomial	Number of defectives in n sample size drawn from a large lot having p fraction defectives; probability of x occurrences in a group of y occurrences, that is, situations involving "go-no-go," "OK-defective," "good-bad" types of observations. Proportion of lot does not change appreciably as a result of sample drawn	Inspection for defectives in a shipment of steel parts; inspection of defective tires in a production lot; determination of defective weld joints; probability of obtaining electrical power of a certain wattage from a source; probability that a production machine will perform its function
Hypergeometric	Inspection of mechanical, electrical, etc., parts from a small lot having known percent defectives. Same as in binomial cases, except the proportion of lot may change as a result of sample drawn	Probability of obtaining 10 satisfactory resistors from a lot of 100 resistors having 2% defectives; similar cases involving light bulbs, piston rings, transistors
Poisson	Situations where the number of times an event occurs can be observed but not the number of times the event does not occur. Applies to events randomly distributed in time.	Number of machine breakdowns in a plant; automobiles arriving simultaneously at an intersection; number of times dust particles found in atmosphere in some number of spot checks; industrial plant personnel injury accidents; dimensional errors in engineering drawings; automotive accidents in a given location per unit time; automotive traffic; hospital emergencies; telephone, circuit traffic; a defect along a long tape, wire, chain, bar, etc.; tire punctures; stones hitting windshield; number of defective rivets in an airplane wing; radioactive decay; number of engine detonations; number of flaws per yard in sheet metal;

The last definition to be introduced is the **mean time between failure (MTBF)** (better: mean operating time between failures). It is widely used to classify reliability of systems and unfortunately misused in many cases. The definition of MTBF for repairable systems with a constant failure rate is:

$$t_{MTBF} = \frac{1}{\lambda} \quad (11)$$

This assumes that the item is as-good-as-new after each repair and λ is constant (exponential distribution) [39]. As we have seen, this assumption holds not true for many practical applications, in which infant mortality and wear-out cause a varying failure rate over the items lifetime. Also, it is important to consider that while many products are very reliable (MTBF of several million hours), their service life can be very short

(for example minutes, in the case of missiles). 25 year old humans have an MTBF of about 800 years [44]. Thus, it is important to not forget that a high MTBF does not necessarily correlate with the length of the service life and that due to variations of λ over time, the MTBF value shall only be used for describing needed repairs in the region of constant failure rate (for example in airplanes).

As Sarsfield [28] noted, spacecraft usually don't follow a constant failure rate over time. In his studies he described that the Weibull distribution seems to fit best for on-orbit spacecraft failures, and that on-orbit failure rates diminish over time. He also described the problem of too pessimistic reliability estimates: relying solely on the constant failure rate approach will cause additional design efforts, biased performance trades and subsequently unwanted cost and schedule growth in most projects [28]. Thus, it is important to study past missions and learn about root causes, risks, and the accuracy of past reliability estimations. We will do that in the following subsections, starting with an overview on risks and causes of failures for space missions.

2.1.2 Classification of Space Missions Failures and their Root Causes

Space is a unique and extreme environment, posing a broad range of different risks for any mission. Effects from the space environment are the most obvious reason for spacecraft failure. Nevertheless, root causes of failure can range from different aspects of spacecraft production, such as manufacturing and design flaws, regular wear-out, or interference by human technological activities to yet unknown causes. Failures can either be randomly distributed over the mission, clustered at the beginning or at the end of the mission.

The **space environment** poses a variety of challenges for satellite hardware. The NASA Reference publication [50] on the topic of space system failure attributed to the space environment listed nine environments that have to be taken into consideration for spaceflight: the neutral thermosphere, the thermal environment, plasma, meteoroids and orbital debris, the solar environment, ionizing radiation, the geomagnetic field, the gravitational field, and the mesosphere. Electromagnetic radiation, charged particles, atomic oxygen, and the extreme thermal environment are, amongst others, reasons for spacecraft to fail in earth orbit. Charged particles can lead to effects such as Total Ionizing Dose (TID), Electrostatic Discharge (ESD), Surface Charging, Internal Charging, Displacement Damage and Single Event Effects (SEEs) [51]. TID results from surface and internal charging due to the bombardment by charged particles. It is an accumulating effect and mostly causes gradual degradation of electronics. ESD is defined as an arc, generated by accumulated charge that goes through material, along surfaces or between components and causes electromagnetic interference. Surface Charging results in arcing and is caused by a buildup of charge on the outer surface of the spacecraft. Internal charging is similar, with the difference that the charge is created in internal components of the spacecraft, also resulting in arcs between circuit boards or other electric components [51]. As TID, ESD and both Surface Charging and Internal Charging are accumulated effects (they build up over time) they could also be seen as wear-out effects. SEEs are caused by impacts of high-energy particles (heavy ions, protons, and neutrons). Their impact results in charge that is greater than the charge carrying an elementary information in the component [52]. This charge can impact electronic components in a broad variety of destructive and non-destructive ways. Destructive errors, also called hard errors, lead to a non-recoverable state. Amongst others, Single Event Latch-up (SEL), Single Event Burnout (SEB), Single Event Gate Rupture (SEGR) and Single Event Hard Error (SHE) are the most important concerns. Depending on the device, there is also a variety of Soft Errors possible: Single Event Upsets (SEU), Single Event Transients (SET), Multiple Cell Upsets (MCU) and Single Event Functional Interrupts (SEFI). A deeper discussion on SEEs and their influence on electronics is provided in [52] and [53]. SEEs occur randomly. Thus, an increased SEE rate would raise the failure rate of all phases in the bathtub curve. According to Hecht [54], all failures originating in the environment of the mission can be diagnosed by the load of the environment exceeding original specification, depicted also in Figure 2-5. As this topic is very broad, programmatic concerns of the effects of the space environment on spacecraft are

summarized in Table 2-3, and the effects of the space environment on different subsystems of spacecraft are depicted in Table 2-4. Both tables are shown at the end of this subsection and follow the NASA Reference Publication on this topic [50].

As we will later see, **design errors** are still a major source of spacecraft failure. Sarsfield classified design failures as failures that occur when the strength of parts is not sufficient to withstand the loads during the mission of the spacecraft [28]. Hecht & Hecht argued that design failures can be diagnosed if repetitive analysis shows that strength is inadequate in some circumstances for the mission [54], as shown in Figure 2-5. It is important to see design failures as failures associated with oversight or error [28]. According to Birolini, such failures should not be considered for infant mortality, since they are already manifesting themselves at $t = 0$ [39]. That is arguable for spaceflight applications, since most spacecraft are not in full operational mode straight from orbit insertion and usually the careful checkout of all subsystems and software will last for days or weeks. Thus, design failures can remain unnoticed and show themselves only later in the mission. That is also true for errors related to software, which we will cover later in this subsection.

Manufacturing errors can be counted as failures arising from a lack of quality. Hecht & Hecht noted that these quality issues might be diagnosed by the variation of strength of parts exceeding the specification [54], as depicted in Figure 2-5. Poor workmanship can lead to such errors in handling and processing. As described before, it is important to consider that, different from terrestrial applications, spacecraft are still manufactured in a very custom way and depend heavily on humans instead of automated processing equipment [28]. Hence, errors due to poor workmanship could be much more prevalent in spacecraft than in modern customer products (which are often produced in automated facilities). Similar to design errors manufacturing errors are already in the system at $t = 0$ and they can also remain unnoted within the spacecraft's life until the function is activated.

Part Failures are the classical failures used by current reliability prediction methods. Sarsfield described that they are linked to the absence of unexpected environmental loads or a clear design error [28]. According to Hecht & Hecht, they have a non-repetitive cause and are only diagnosed if there is no other cause likely [54]. It is important to differentiate between part failures in the constant region due to non-repetitive causes, and **part failures due to wear-out**. Non-repetitive part failures occur randomly and with a constant failure rate. Thus, they are one reason for the flat middle region within the bathtub curve. Wear-out has a repetitive cause, no matter if it is due to degradation of strength, accumulating effects of space radiation, or wear and tear in case of rotational devices such as reaction wheels. Wear-out is not randomly distributed, it is described by the third zone of the bathtub curve and occurs with an increasing failure rate. On the other hand, Reeves [42] identified six reasons that can lead to failures on the part level. Besides being worn out, initially defect or damaged by poor workmanship, he further described improper application, drifting out of initial settings due to gradual degradation, and part failure as consequence of other parts failing as those reasons. Since this mixes up different root causes, we will define part failures in this work as the ones with a non-repetitive cause, thus being randomly distributed over time. All other failures will be named after their root cause (workmanship, design, manufacturing) or their physical root cause (wear-out = degradation of physical parameters due to the environment or usage). Figure 2-5, adapted from Hecht & Hecht [54], shows the major failure mechanisms for spacecraft.

Apart from these failure mechanisms, other root causes exist for failure: **Operational errors**, which are incorrect commands leading to abnormal behavior or failing to take action when needed, are examples of human interference leading to spacecraft failure. Other examples for that would be jamming, both unintentional or intentional, and cyberattacks on a satellite system's space or ground segment [51].

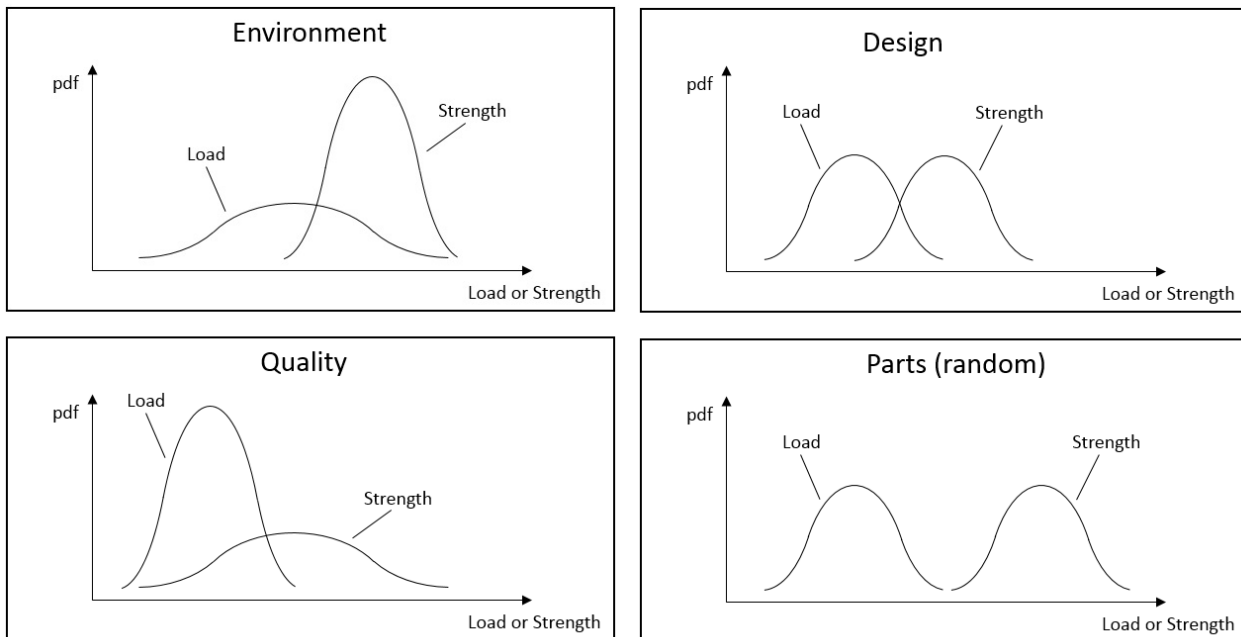


Figure 2-5: Failure Mechanisms in Spacecraft. Adapted from [54]

It is important to note that the failure rate of electronic systems is traditionally determined by adding up the failure rates of the parts of the system (bearing in mind the redundancies, in some cases). But as Goel & Graves noted [40], increased system complexity and improved component quality have shifted the root causes of failure away from component to system-level factors, including manufacturing and design. According to Sarsfield [28], design and environment causes are the most significant sources of failure in space systems. A 1994 study of planetary spacecraft included in Sarsfield's report noted that 60% of the failures that occurred during test and integration could be traced back to design problems.

An increasing cause of error on spacecraft are **software failures**. Cheng reported that over half of all failures in spaceflight¹⁴ between 1998 and 2000 involved software. For software, small errors can be fatal, as redundancy is ineffective and even more human factors are involved as in hardware [55]. Losses of high-asset missions such as Mars Climate Orbiter [56], Ariane 5 flight 501 [57] and Mars Polar Lander [58] can be directly traced back to software flaws. As Lowry showed, a culture within the Ariane program existed of only addressing random hardware failures¹⁵ and duplicate backup systems were put in place as failure handling mechanisms. However, in case of software failures, which are essentially design errors, failure of the primary system highly correlates with failure of the backup system [59]. For Ariane 501, the design of the main computer of the inertial guidance system was built in a way that it shuts itself down in case of exception in an unnecessary function. As the alignment function created such an exception after liftoff, the primary and backup system shut down just the way they were designed to, leading to loss of the rocket [60]. Again, this design would have been sufficient in the case of purely random hardware failures. But software tends to cause system failures, also called component interaction accidents, thus failures that result from dysfunctional interactions among components and not from failure of a specific part or component [60], [61].

Software complexity, measured in source line of codes (SLOC), has increased steadily over the last decades, similar to the increase of electronic parts shown in Figure 2-1. In its first flight in 1969 the Boeing 747 airplane worked with approximately 400 KSLOC¹⁶. The Boeing 787 airplane, quite similar in size,

¹⁴ Including launch vehicles.

¹⁵ We will come back to the (mis)use of random hardware failures for reliability prediction in Section 2.2.

¹⁶ 1 KSLOC = One Thousand SLOC.

experienced a growth to approximately 13 MSLOC¹⁷ in 2009 [62]. Space systems software, similar to that, experienced an exponential growth with a factor of 10 every 10 years [63]. Tosney and Pavlica [37] also noted this exponential rate of software growth in spacecraft over the last decades (see Figure 2-6).

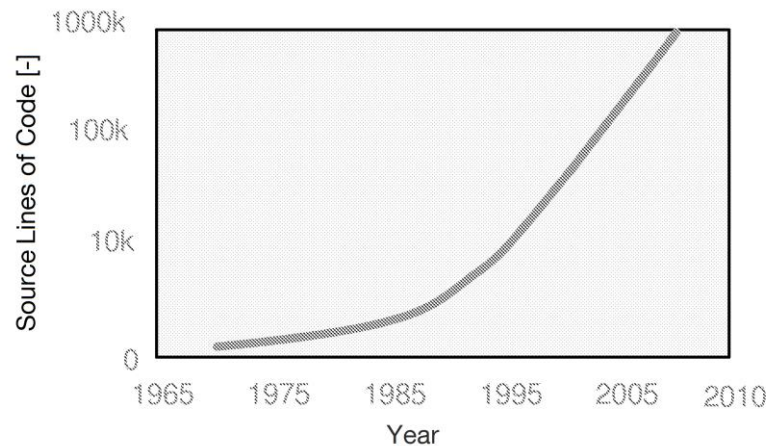


Figure 2-6: Exponential SLOC growth in spaceflight projects. Adapted from: [37]

Interestingly, new-designed systems experience a larger growth of software than traditional systems or follow-on missions [64]. This is important, since growth of on-orbit and also pre-launch anomalies can be directly correlated to growth of SLOC [63], [64]. Due to their limited resources, CubeSats and small satellites traditionally shift functionality from hardware to software and many of them can be categorized into the “newly-designed system” group. Deep Space missions, with extensive management overhead, reach around 40% probability of a critical software error at 100 KSLOC. That means there is a low chance of a deep space mission being free of critical software errors beyond 200 KSLOC [59]. It is questionable whether low-cost missions reach the same quality of software development as deep space missions. Interestingly, terrestrial applications such as cars work reliably while reaching around 100 MSLOC in 2010 [63]. The difference between the two is that the first deployment of the product in terrestrial applications is almost never free of critical errors [59], but also not expected to be so. After the first deployment of the terrestrial product, extensive beta testing is used to debug on the system level, sometimes even involving selected groups of consumers to further enhance test heterogeneity.

We initially defined software failures broadly as design failures. Depending on their characteristics, a further distinction is possible [65]: so-called Bohrbugs, deterministic bugs, which are easy to isolate, manifest themselves consistently under well-defined conditions, are the first group of software failures. So-called Mandelbugs, the second group of software failures, are non-deterministic and thus difficult to isolate and reproduce. A further distinction can be made between non-aging related Mandelbugs and those related to software aging [65]. While aging and wear-out is normally not associated with software, error accumulation in internal states can lead to this type of failure [66]. Of course, this could also be seen as a software design error since such accumulations would have to be prevented by design. Grottke, Nikora & Trivedi [67] studied 18 JPL/NASA space missions and classified the identified software faults into the following four categories. Of the 520 software faults found¹⁸, 319 were Bohrbugs, 167 non-aging related Mandelbugs, 23 aging-related Mandelbugs and 11 could not be determined [67]. This means that roughly 2/3 of the faults in the software of these high-asset missions were due to deterministic bugs, which would have been relatively easy to detect. In a later paper, Alonso, Grottke, Nikora & Trivedi [65] found that aging-related Bohrbugs experience a decreasing failure rate, while non-aging related Mandelbugs are best modelled by a constant

¹⁷ 1 MSLOC = One Million SLOC.

¹⁸ Overall 1300 anomalies were found, 25% were identified as software anomalies.

failure rate. The same researchers also concluded that the majority of Bohrbugs (over 75%) is solved by applying a fix within the mission. Fixes are also the most frequent type of mitigation action taken for the other two types of bugs, although “use as is” is also quite common¹⁹ [68]. For small satellites and CubeSats, these fixes are often harder to accomplish since functionality on the satellite but also manpower on the ground is limited. Also, the communication to the satellite is in most cases more restricted than in traditional high-asset missions. For traditional missions, communication to the satellite can be maintained 24/7 and thus many parameters of the satellite can be traced back. On-orbit data can be applied on-ground support equipment or engineering models, and root causes of software bugs can be determined. In most of the cases that is not possible for small satellite missions. Therefore, although it can be assumed that software bugs, Bohrbugs in majority, will also happen more frequently in the future on small missions, traditional ways of coping with that are only of limited applicability. Taking one final look at the software growth in space applications, Judas & Prokop reported a mathematical function in 2011 [62] to predict the SLOC of future unmanned missions. For their fit, they studied the SLOC of spacecraft launched between 1962 and 2008 and values ranged from 30 SLOC for one Mariner mission to 1 MSLOC for the Jules Verne ATV spacecraft. The correlation was relatively weak ($R = 0.667$) but showed an exponential growth over time similar to the other studies.

So, what are the most important considerations while dealing with growing software in spaceflight? Flight software per mission as well as number of problems associated to flight software per mission have grown steadily over the past four decades. Complexity in flight software enabled new functionality and progress in space missions but also increased the overall risk. New functionality is more often added in software or firmware than in hardware [63]. This also happens in small satellites, in which limited resources and the small envelope often result in a preference of software over hardware solutions. Although many small satellites utilize COTS from terrestrial applications, there are differences between commercial software used in most terrestrial applications and space software. We have to interact with the satellite using a radio link that is quite different from most terrestrial interfaces. The radio link is often limited – thus the information coming back from the system is limited too. Flight software has to coordinate multiple devices, sometimes with timing constraints, monitor and control them in a coordinated way, while dealing also with the harsh environment of outer space. Traditionally, flight software runs on limited resources due to the usage of slow, rad-hard processors [63]. Small satellite software also must operate on limited resources, although for a different reason. As Leveson noted [60], software and digital systems require changes of engineering practice, as they do not fail in random behavior as expected by many traditional approaches. Failure mode and effects analysis (FMEA), fault tree analysis (FTA), overdesign, safety margins and redundancy are not very effective against software failures, as software failures originate from design flaws. Although “diversity”, i.e., having multiple versions of software written by different programmers, is used to cope with software errors through redundancy, it has been shown that this approach is not valid in praxis, as the different versions will not fail in a statistically independent way²⁰ [61]. As we have seen, software problems often originate from component interactions and not from component failure itself. In most cases Leveson analyzed, the software and hardware components acted according to their specification. It was the combined behavior that led to system failure. Flexibility of software and the increasing number of interfaces create systems, in which the interactions among the components cannot be fully planned, understood, anticipated or guarded against. Adding redundancy increases complexity, and thus intensifies the problem²¹. Leveson further noted that there have been examples of systems failing due to introduced

¹⁹ We will have a more detailed look into these statistics later in this subsection.

²⁰ Which is not surprising, as human designers do not make random mistakes, or as Leveson puts it, human developers are not just “monkeys typing on typewriters”. Often the same common design error is likely to happen with different people or groups of developers [61].

²¹ Agreeing with the statement by Fleeter (Equation 1) in Section 1.3.

redundancy [61]. It is therefore important to understand that software faults are latent design errors, and the complexity of today's spacecraft will increase the chance that they emerge late in the process.

Before we will have a detailed look into past analysis of time-dependent on-orbit reliability data, the final part of this subsection will deal with recent assessments of the causes for spacecraft failure. A report by the Aerospace Company in 2009 [23] reviewed 325 space vehicle anomalies and grouped them into random hardware faults, systematic faults and faults in which the cause remained unknown. Random hardware faults describe what we have called part failure, and is assumed to occur with a constant failure rate. Systematic faults include software faults, engineering faults (occurring in parts, material, and design), workmanship faults, limited engineering knowledge (meaning failure of first attempt at new phenomenology), launch vehicle failure, and faults due to the space environment. Systematic faults may affect more than one component [23], and cannot be assumed to occur with a constant failure rate²². The report classified 73% of the anomalies as systematic faults, 16% as random hardware faults, and 11% as faults in which the cause remained unknown. Of the systematic faults, 33% were due to workmanship issues and 30% due to engineering faults. Figure 2-7 depicts the detailed breakdown of the causes. Nieberding [69] reported that out of 39 cases analyzed only one had a random part failure as the cause of error. He further stated that history demonstrates that tests produce unexpected and unwanted results and that a zero-based test approach wrongly assumes that spacecraft designers can foresee every potential aspect of the system under all conditions [69]. As we have learned, this is clearly not the case in our complex, interdependent space systems. Finally, a presentation given by the European Space Agency (ESA) in 2016 [70] showed results of on-orbit feedback of the fleet of Airbus Defence and Space (former EADS Astrium) satellites. Although the fraction of issues due to the space environment is higher than in the report by the Aerospace Company²³, systematic issues due to design, manufacturing and operations is the biggest fraction of causes for errors²⁴. This breakdown of causes is shown in Figure 2-8. It was further reported by ESA that in current reliability prediction methods a significant amount of on-orbit anomalies due to non-random failures (design, manufacturing, workmanship) is not covered [70]. We will discuss that in Section 2.2.

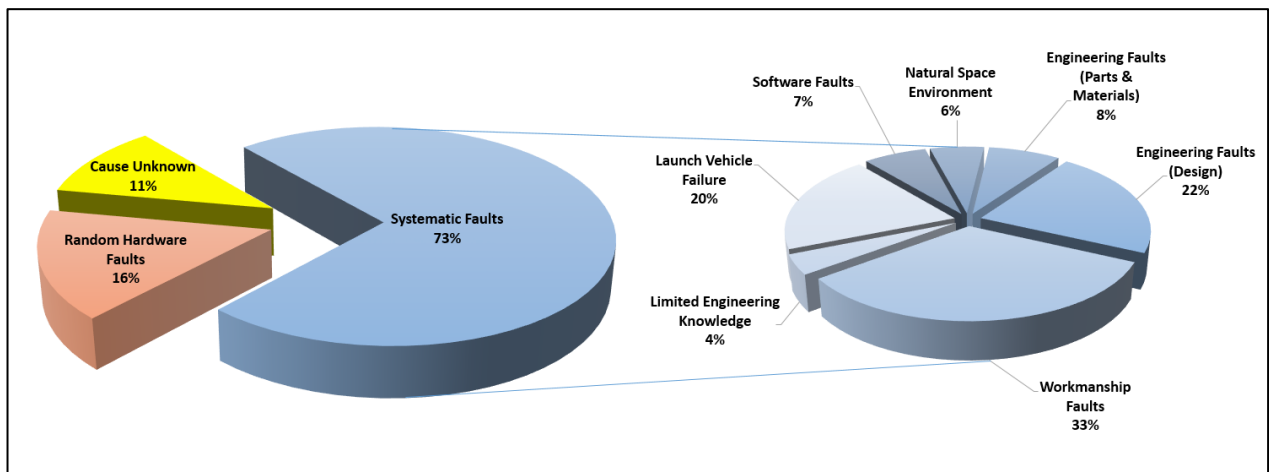


Figure 2-7: Breakdown of the causes of spacecraft faults analyzed by the Aerospace Company. Adapted from: [23]

²² As we have seen, this is arguable for SEEs.

²³ This could be partly due to the different operating environment (i.e., geosynchronous orbit) for Airbus's satellites.

²⁴ Unfortunately, the breakdown also comprises another not further specified cause: „failures“.

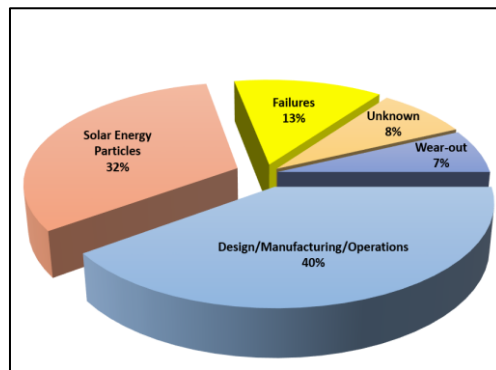


Figure 2-8: Breakdown of causes for anomalies on satellites by EADS Astrium (2012). Adapted from: [70]

As Wertz [30] noted, current space systems primarily rely on high reliability on part level, but parts are not the typical reason why missions fail. By adding more complexity, more software and redundancy, we often worsen the overall system reliability, sometimes leading to a reliability of zero in case the satellite is not launched in time or cancelled, as already pointed out. We will later assess the flawed assumption of “high cost parts equal high reliability”, and study if the flight proven label (i.e., heritage) is really a one-for-all solution for spacecraft development. As Sarsfield [28] noted, some equipment will be re-used to applications not intended by the original designer, which was one of the main causes for the loss of the Mars Observer spacecraft in 1993, in which hardware from LEO missions was wrongly used on an interplanetary mission. A failure investigation board by NASA for that mission concluded that additional ground testing would have revealed many problems associated with the re-use of equipment in that mission [28]. Before we come to a critical assessment of testing efforts in space missions in Section 2.2, we will first have a look on historical spacecraft reliability and on-orbit failure studies in the next subsection.

Table 2-3: Programmatic concerns of the effects of the space environment on spacecraft. Source: [50]

	DEFINITION	PROGRAMMATIC ISSUES	MODELS/DATABASES
NEUTRAL THERMOSPHERE	Atmospheric density, Density variations, Atmospheric composition (Atomic Oxygen), Winds	GN&C system design, Materials degradation/ surface erosion (atomic oxygen fluences), Drag/decay, S/C lifetime, Collision avoidance, Sensor pointing, Experiment design, Orbital positional errors, Tracking loss	Jacchia/MET, MSIS, LIFTIM, upper atmospheric wind models
THERMAL ENVIRONMENT	Solar radiation (albedo and OLR variations), Radiative transfer, Atmospheric transmittance	Passive and active thermal control system design, Radiator sizing/material selection, Power allocation, Solar array design	ERBE database, ERB database, NIMBUS database ISSCP database, Climate models, General Circulation Models (GCM's)
PLASMA	Ionospheric plasma, Auroral plasma, Magnetospheric plasma	EMI, S/C power systems design, Material determination, S/C heating, S/C charging/arcing	International Reference Ionosphere Models, NASCAP/LEO, NASCAP/GEO, POLAR
METEORIODS AND ORBITAL DEBRIS	M/OD flux, Size distribution, Mass distribution, Velocity distribution, Directionality	Collision avoidance, Crew survivability, Secondary ejecta effects, Structural design/shielding, Materials/solar panel deterioration	Flux models
SOLAR ENVIRONMENT	Solar physics and dynamics, Geomagnetic storms, Solar activity predictions, Solar/geomagnetic indices, Solar constant, Solar spectrum	Solar prediction, Lifetime/drag assessments, Reentry loads/heating, Input for other models, Contingency operations	MSFC EL Laboratory model, NOAA prediction data, Statistical models, Solar database
IONIZING RADIATION	Trapped proton/electron radiation, Galactic cosmic rays (GCR's), Solar particle events	Radiation levels, Electronics/parts dose, Electronics/single event upset, Materials dose levels, Human dose levels	CREME, AE-8MIN, AE-8MAX, AP-8MIN, AP-8MAX, Radbelt, Solpro, SHIELDDOSE
MAGNETIC FIELD	Natural magnetic field	Induced currents in large structures, Locating South Atlantic Anomaly, Location of radiation belts	IGRF85, IGRF91
GRAVITATIONAL FIELD	Natural gravitational field	Orbital mechanics/tracking	GEM-T1, GEM-T2
MESOSPHERE	Atmospheric density, Density variations, Winds	Re-entry, Materials selection, Tether experiment design	Earth-GRAM 95, UARS database, Mars-GRAM 3.34

Table 2-4: Space environment effects on spacecraft subsystems. Adapted from [50].

	SPACE ENVIRONMENTS				
SPACECRAFT SUBSYSTEMS	Neutral Thermosphere	Thermal Environment	Plasma	Meteoroids/Orbital Debris	
Avionics		Thermal Design	Upsets due to EMI from Arcing, S/C Charging	EMI Due to Impacts	
Electrical Power	Degradation of Solar Array Performance	Solar Array Designs, Power Allocations, Power System Performance	Shift in Floating Potential, Current Losses, Reattraction of Contaminants	Damage to Solar Cells	
GN&C/Pointing	Overall GN&C/Pointing System Design		Torques due to Induced Potential	Collision Avoidance	
Materials	Materials Selection, Material Degradation	Material Selection	Arcing, Sputtering, Contamination Effects on Surface Properties	Degradation of Surface Optical Properties	
Optics	S/C Glow, Interference with Sensors	Influences Optical Design	Reattraction of Contaminants, Change in Surface Optical Properties	Degredation of Surface Optical Properties	
Propulsion	Drag Makeup/Fuel Requirement		Shift in Floating Potential Due to Thruster Firings Making Contact with the Plasma	Collision Avoidance, Additional Shielding Increases Fuel Requirement, Rupture of Pressurized Tanks	
Structures		Influences Placement of Thermally Sensitive Surfaces, Fatigue, Thermally Induced Vibrations	Mass Loss From Arcing and Sputtering, Structural Size Influences S/C Charging Effects	Structural Damage, Shielding Designs, Overall S/C Weight, Crew Survivability	
Telemetry, Tracking, & Communications	Possible Tracking Errors, Possible Tracking Loss		EMI Due to Arcing	EMI Due to Impacts	
Thermal Control	Reentry Loads/Heating, Surface Degradation due to Atomic Oxygen	Passive and Active Thermal Control System Design, Radiator Sizing, Freezing Points	Reattraction of Contaminants, Change in absorptance/emittance properties	Change in Thermal/Optical Properties	
Mission Operations	Reboost Timelines, S/C Lifetime Assessment	Influences Mission Planning/ Sequencing	Servicing (EVA) Timelines	Crew Survivability	
SPACECRAFT SUBSYSTEMS	Solar Environment	Ionizing Radiation	Magnetic Field	Gravitational Field	Mesosphere
Avionics	Thermal Design	Degradation: SEU's, Bit Errors, Bit Switching	Induced Potential Effects		
Electrical Power	Solar Array Designs, Power Allocations	Decrease in Solar Cell Output	Induced Potential Effects		
GN&C/Pointing	Influences Density and Drag, Drives Neutrals, Induces Gravity Gradient Torques		Sizing of Magnetic Torquers	Stability & Control, Gravitational Torques	Effect on GN&C for Re-entry
Materials	Solar UV Exposure Needed for Material Selection	Degradation of Materials			Degradation of Materials Due to Atmospheric Interactions
Optics	Necessary Data for Optical Designs	Darkening of Windows and Fiber Optics			
Propulsion	Influences Density and Drag			Influences Fuel Consumption Rates	
Structures	Influences Placement of Thermal Sensitive Structures		Induces Currents in Large Structures	Propellant Budget	Tether Structural Design
Telemetry, Tracking, & Communications	Tracking Accuracy, Influences Density and Drag		Locating South Atlantic Anomaly	May Induce Tracking Errors	
Thermal Control	Influences Reentry Thermal Loads/Heating				
Mission Operations	Mission Timelines, Mission Planning	Crew Replacement Timelines			

2.1.3 Reliability Analysis of Satellites

As shown in the last subsection, reliability of spacecraft has some unique characteristics different from terrestrial applications. The complexity of novel space applications often leads to engineering errors, resulting in early failures. Also, design parameters such as the solar array size of satellites, and expendables such as fuel, must be carefully chosen for the planned end of life of the satellites since carrying excess mass into orbit directly increases mission cost. Historically, some missions have continued to operate well beyond their planned mission lifetime and justified the additional expendables, while others heavily underperformed. To better understand reliability engineering decisions for spacecraft and the underlying reasons we will take a closer look at the historical reliability data of spacecraft and the reliability analysis methods used in this subsection. We will move through the subsection mostly in a historical sequence, but will conclude by summarizing the work of Saleh et al. [22], which is by far the most extensive research on the reliability of spacecraft within the last decade.

Glennan, the first administrator of NASA, started coordinated reliability efforts for space in 1959. The work was motivated by examples from the automotive industry, in which the part quality level steadily rose since the invention of cars²⁵. Estimates for the Mercury program showed that the Atlas rocket had about 40,000 critical parts involved and that this could double when putting a Mercury capsule on top of the rocket. Thus, a reliability program for Mercury needed dedicated groups of reliability engineers to disentangle this complex task. Terms such as “critical part“, “system” and “failure” had to be revisited and defined for their use in reliability engineering at NASA. Also, systems engineers had to define how to measure overall system performance from subsystem data and what kind of indices or coefficients to use for that task [71].

The first feedback from orbit came in the early 1960’s when Willard [72] first analyzed the on-orbit reliability of six satellites²⁶ with a cut-off at April 30, 1961. He reported that the observed values for on-orbit reliability were 5-10 times better than the predicted ones. Both the estimated and the predicted life was in the hundreds or thousands of hours, which was relatively short. Two approaches were used for reliability prediction: A Part Failure Rate Method, applying constant failure rate on each part of the system, and a second method, in which the system was segmented in active element groups of two different stress levels [72]. In 1965, Bloomquist et al. [73] and later in the same year the Planning Research Corporation [74] reported on the reliability assessment of the Geodetic Earth Orbiting Satellite (GEOS)-A spacecraft. Using the MIL-HDBK-217F for reliability prediction, they showed a figure of merit of only 0.41 for the accumulation of the mission’s value in the first year, assuming all experiments onboard the spacecraft would have a reliability close to 1 within that timeframe. This estimation assumed a reliability of the spacecraft’s main units between 0.7 (command system) and 0.932 (main battery) [74]²⁷.

In 1966, Timmins [75] first reported on the effectiveness of system tests for achieving reliable satellite performance. For that, he looked at the laboratory results of 64 spacecraft tested at Goddard Space Flight Center (GSFC) and at the space performance of the first 10 of these spacecraft. He emphasized the importance of prototype models to detect problems with design, quality control, and materials and suggested to evaluate operating procedures early in the program. 216 problems (75 major, 134 minor) were detected on seven prototype spacecraft in the system tests and 97 further problems were detected in the system tests of ten flight spacecraft (two catastrophic, 43 major, 52 minor). At the same time, he also reported on the effectiveness of tests under simulated environment to detect workmanship problems and

²⁵ Swenson, Grimwood and Alexander present a nice example for that: the quality between a 1927 car and a 1959 car had to rise, since the involved critical parts of cars approximately tripled during that time. Thus, a low amount of parts made a limited reliability of those parts acceptable in 1927 cars, but not in 1959 cars. Therefore, the more critical parts there are in a system, the higher the quality level of each part has to be [71].

²⁶ Explorer VII, Tiros I, Transit IB, Transit IIA, Transit IIB and Courier IB [72].

²⁷ The 175 kg GEOS A spacecraft was launched on November 6, 1965 and operated until January 1967, when the satellite’s command system failed.

early failures, which can also be used to assure an acceptable level of confidence in the system in order to launch it. Figure 2-9 shows the saturation curve of failures over testing time of seven spacecraft tested in simulated space environment. Timmins considered the time to reach the plateau as the minimum testing time needed in simulated space environment. This is the first source for the use of saturation curves in spaceflight testing known to the author of this thesis. The 10 spacecraft launched showed clear signs of infant mortality, with the experiments being the most vulnerable subsystem. Timmins argued that this is not surprising since state-of-the-art hardware had to be used for most of the experiments to achieve their mass, size, and functionality for space applications. To achieve better reliability in the future, he recommended to use better procurement specifications, high reliability parts, inspection, and other quality assurance provisions [75].

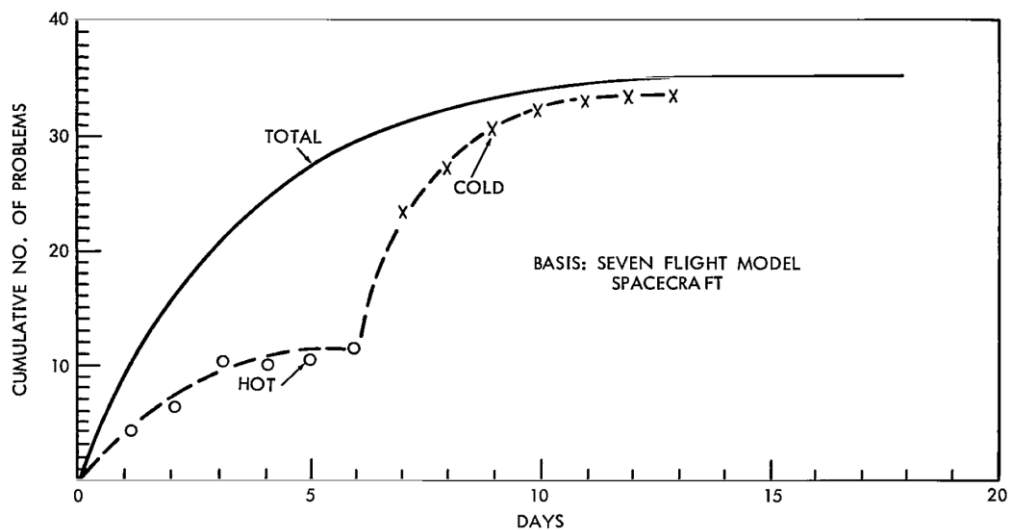


Figure 2-9: Saturation curve system level testing of seven flight model spacecraft in simulated space environment. Source: [75]

In the same year, Mercy [76] also reported on the contribution of environmental test to spacecraft reliability. He studied 49 launches with 251 experiments from 1960 to 1966 and found that 63% achieved their mission objective as planned, with further 30% obtaining useful data. Through his data, Mercy emphasized the necessity of full-scale system testing of prototype and flight spacecraft. Also in 1967, Wright [77] first reported on failure rate computations for interplanetary spacecraft. He studied 1,500 problem and failure reports of the Mariner IV mission, using the JPL problem and failure reporting system. He compared flight results of Mariner IV (no known relevant failures) with test results of this spacecraft (11 relevant failures), concluding that once a spacecraft is past the launch environment, the probability of mission failure is greatly lowered. For Mariner IV, the dominant failure experience period occurred during final subsystem testing and full level system testing, since failure caused by design, workmanship, and human error were revealed in that period²⁸ [77].

Also in 1967, New & Timmins [31] published work on the effectiveness of environment simulation testing for spacecraft. In an approach similar to earlier studies, they evaluated the GSFC test philosophy on 16 prototype units and 48 flight units, and reported 855 problems uncovered and corrected during testing [31]. At the end of the 1960s, Boeckel & Timmins [78] presented research on the test plan optimization of explorer-size spacecraft. Due to their size, this class of spacecraft is especially interesting for our studies. They developed a mathematical cost model, and reported on the failure probability of subsystems and data gained from system tests of six Interplanetary Monitoring Platform (IMP) satellites. The reliability of the subsystems on this platform was between 0.5 and 0.9 [78].

²⁸ Mariner IV later performed the first successful flyby of the planet Mars in July 1965.

Boeckel, Timmins & Mercy [32] continued in 1970, reporting a 95% successful space record of spacecraft tested at GSFC. In their report they showed that the specific characteristics of spacecraft, namely being one-shot and also one-of-a-kind systems in most cases, prevent the usage of statistical approaches developed for mass production. They stipulated that to assure a high probability of success, the actual flight hardware must be exposed to a simulated space environment before launch. System tests, despite rigid quality assurance requirements, proved to be indispensable, and especially complex spacecraft experienced high numbers of malfunctions during system level tests. In 22 system tests of flight model spacecraft, 759 malfunctions were detected and 46% of these malfunctions were found in system level functional testing, 42% while testing the system in a simulated space environment, and 12% in structural tests. Similar to the Timmins report of 1966 [75], experiments proved to be the most unreliable subsystem (51% of total malfunctions). Boeckel et al. [32] also used saturation curves in thermal-vacuum tests to show the minimum test period needed (16 days) to reach a plateau for failures over time. They argued that with sufficient environmental testing before launch, the random failure rate region of the bathtub curve can be reached, and on-orbit reliability enhanced. In their study, they showed that compared to spacecraft not tested at GSFC, their own system environmental tests were able to reduce, but not completely eliminate, infant mortality. Overall, 46% of the 24 studied spacecraft experienced a failure on the first day, and 20% of all recorded failures occurred on the first day on-orbit. Furthermore, 63% of all spacecraft had failures in the first 30 days, and 35% of all failure concentrated in that time period. Nevertheless, they reported that most spacecraft have later outlasted their intended lifetime by a significant period [32].

Since early failures proved to be higher than expected, Timmins & Heuser [79] specifically researched first-day failures in 1971. They studied 57 spacecraft and found 69 malfunctions in this group. 13% of these malfunctions had catastrophic or major degrading consequences, and they assessed the origin of these failures mostly to design problems that were not or could not be tested adequately. Again, the largest proportion (50%) of the malfunctions occurred on experiments. Nevertheless, according to Timmins & Heuser, 56 of the 57 spacecraft returned useful scientific data. Of the 57 spacecraft studied, 27 had no first-day malfunction at all and redundancy eliminated 20% of the malfunctions occurring on the remaining other spacecraft [79]. Timmins continued their research with a study on first month spacecraft performance [80] and total life performance [81] of the same group of spacecraft in 1974 and 1975. The updated findings of both studies are depicted in Figure 2-10. Of 57 spacecraft, 45 had one or more than one malfunction within the first month. A total of 157 malfunctions was reported in the 1974 study, whereof only 5% resulted in major loss of functionality (50-100%), mainly due to redundancy. 50% of the failures of the first month happened on the first day of the mission [80].

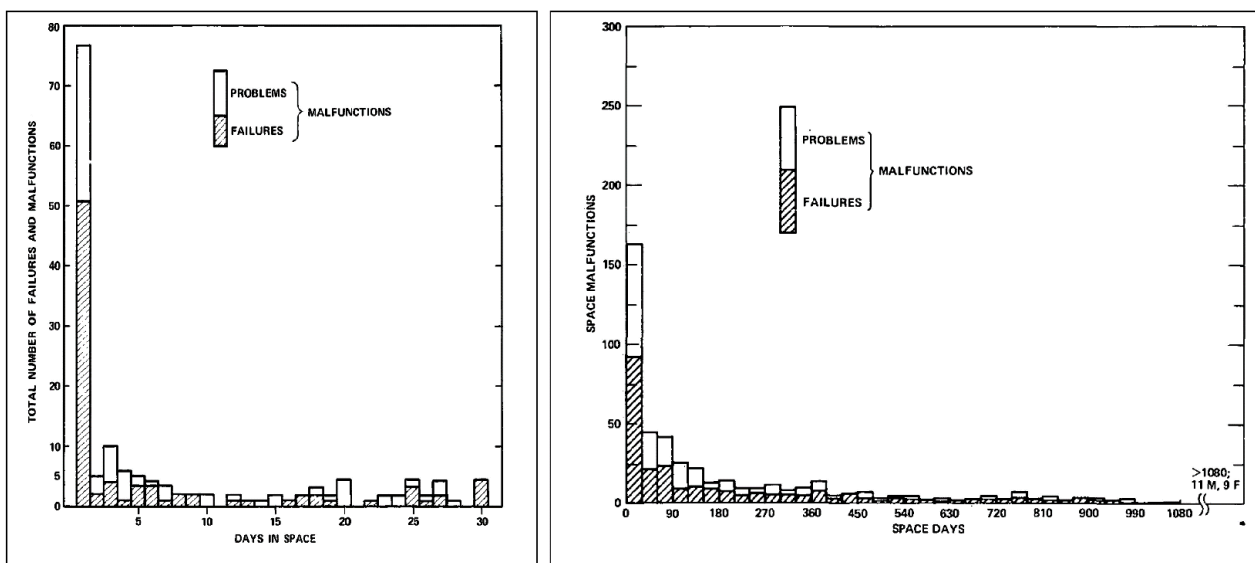


Figure 2-10: First month and total life on-orbit malfunctions of 57 spacecraft. Adapted from [80] and [81].

The root cause for more than half of the failures (64%) as well as overall malfunctions (59%) were in the electrical domain. Consistent to other work, Timmins identified the experiment subsystem as the main source for failures (60%) and malfunctions (47%), with command and data handling was the second biggest source (14% for failures and 20% for malfunctions). He continued analyzing which minimum level of hardware assembly would have been necessary to detect the malfunctions. For almost half of all cases, system level testing would have been necessary to detect the error (system level 49%, subsystem level 29%, component level 18%, below component level 4%). He concluded his study with an estimation of the proportion between failures found in system level tests to failures occurring within the first month in space. For most devices, 5-10 times as many failures were found in system tests on-ground as later occurred within the first month in space [80].

The 1975 study of Timmins [81] reported a total number of 449 malfunctions over three years of data coverage. 85% of these malfunctions were classified as minor loss of function, which means that they caused less than 10% of loss of functionality for the mission. For the first time to the knowledge of the author, a time-dependent distribution of spacecraft alive after launch were given (see Figure 2-11). Timmins discussed that the large number of malfunctions and failures in the first 30 days on-orbit shows that the infant mortality type of errors was not completely removed before launch. He further continued that if there is a constant failure rate region it does not occur until 90 days or more on-orbit. Timmins found no indication for wear-out in the studied spacecraft within the three years' timeframe [81].

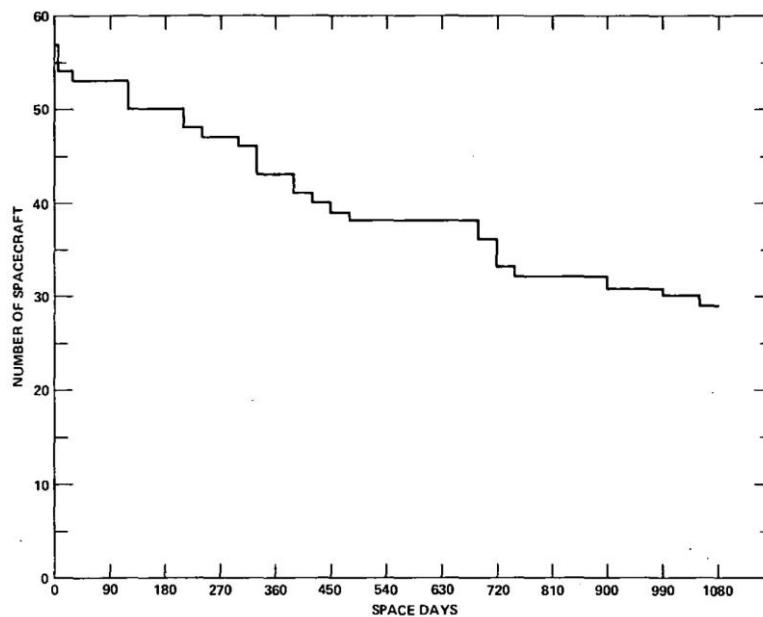


Figure 2-11: Number of spacecraft alive at a certain day after launch. Source: [81].

Timmins concluded his study with an overall rate of failures per spacecraft for environmental tests, for the first day in space, for the first month in space and for the total life. Had there been an average of 12 failures on the spacecraft in system environmental level testing, the number of failures was reduced to an on-orbit average of 0.9 failures per spacecraft on the first day, 1.7 within the first 30 days, and 5.0 over the total life. He also hypothesized what would have happened if the system level test program had not taken place: the total space malfunctions would have increased by a factor of four, with 75% of them occurring within the first 30 days in space. That would have resulted in an average of 12 failures per spacecraft within the first 30 days [81]. These results are also supported by an earlier study on flight model spacecraft performance during thermal-vacuum tests by Timmins, Heuser & Strain [82]. In a 1973 NASA report they showed test and flight data of 39 spacecraft flight-models. Just as in the 1966 study, they found a plateau in cumulative number of failures vs. test days conducted and subdivided the data in ambient, transient, cold, and hot test

cases. Figure 2-12 shows the saturation curve of failures vs. conducted test days. Timmins et al. noted that the occurrence of failures in thermal-vacuum is associated to time and temperature-stress, not time alone [82].

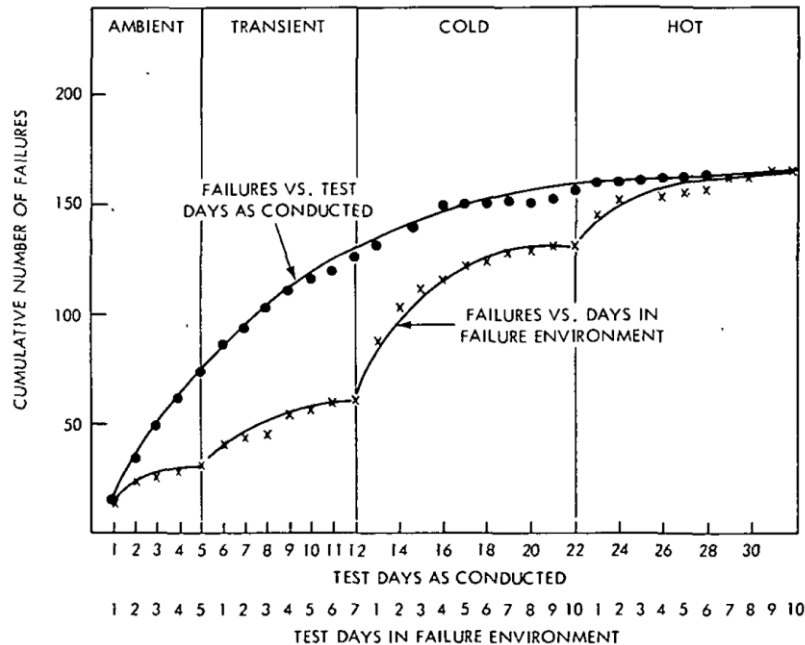


Figure 2-12: Saturation curve of the cumulative number of failures vs. test days of 39 spacecraft flight models. Source: [82]

In 1976, Norris and Timmins [83] reported Duane and Weibull reliability growth models on the 57 spacecraft also used in their earlier research. In their report they also provided failure rates and compared experience from the thermal-vacuum system tests to in-flight performance. The on-orbit failure data showed that the used reliability prediction, based on the number and failure rates of parts, and assuming a constant failure rate, was not adequate. The on-orbit failure rate was better approximated by Duane and Weibull fits, which we will discuss in detail in Section 2.3. Duane and Weibull functions both captured the decreasing failure rate experiences on-orbit, apart from underestimating day-1 failure rates by a factor of four. A location parameter of $\gamma = 3$ days was used to modify the Weibull function. Figure 2-13 (left) shows the Weibull fit curve of failures on-orbit. Figure 2-13 (right) depicts the saturation curves of failures in system level thermal-vacuum testing and the subsequent first 36 days in space. It can be noted that the saturation experienced through system level testing lead to an acceptable failure rate on-orbit. Moreover, the increased first day failure rate, inconsistent with the growth curves, can be seen.

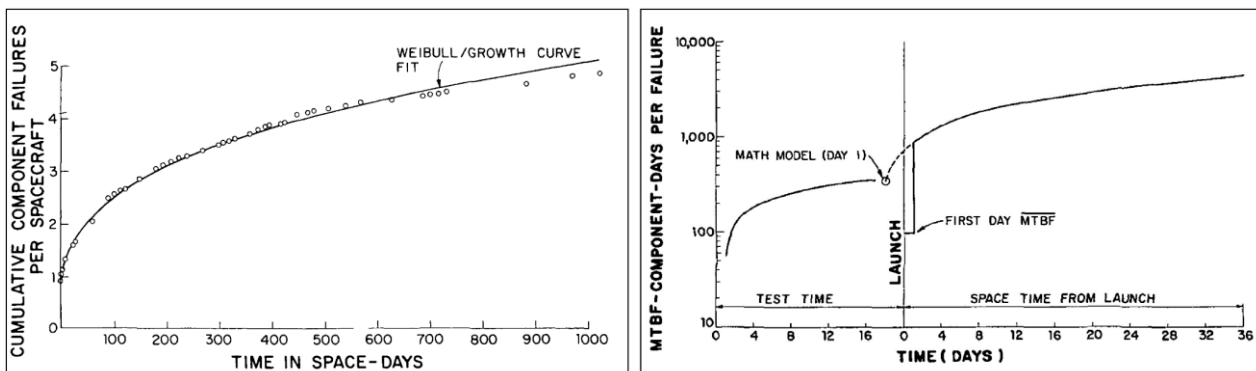


Figure 2-13: Weibull growth curve of on-orbit failures per spacecraft (left) and saturation curves of MTBF over system level thermal-vacuum tests and subsequent time on-orbit (right). Adapted from [83].

In general, Norris and Timmins substantiated the doubts on the constant failure rate model with on-orbit data for the first time. To the knowledge of the author of this thesis they also provided one of the first parametric models for on-orbit reliability, based on their reliability growth function. Figure 2-14 shows the differences between the constant failure rate model and the reliability growth model. Though they gave no further insight into the reason for factor of four elevated first-day failure rates, they stated that many of those first day failures did not result in loss of the spacecraft [83].

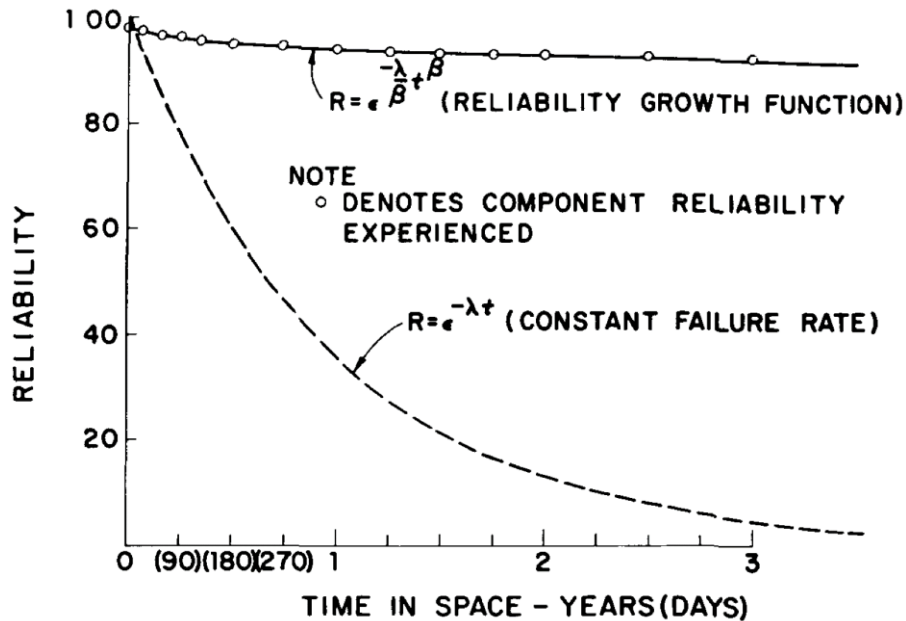


Figure 2-14: Component reliability of 57 GSFC missions and the difference to constant failure rate-based prediction models. Source: [83].

The Planning Resource Corporation's Space Data Bank was used by Bloomquist & Graham [84] to analyze spacecraft anomalies on 316 spacecraft from 1960 to 1975. They were able to categorize over 1,600 separate anomalies into 30 categories, and identified 22 of them for which the then planned Space Shuttle could be utilized for in-space testing. The five categories with the most anomalies were scientific instruments (12.3%), tape recorders (10%), camera equipment (7.3%), batteries (5.5%) and radio frequency (RF)/electromagnetic interference (EMI) (5.3%) [84]. In 1977, Levine [85] reported findings by the Aerospace Corporation on longevity of spacecraft. They analyzed three different sets of data in terms of trends in communication satellites, achieved on-orbit operation lifetime, and predicted versus achieved lifetime. A trend towards growing mass, power and complexity with increasing launch capability²⁹ was reported, as can be seen in Figure 2-15.

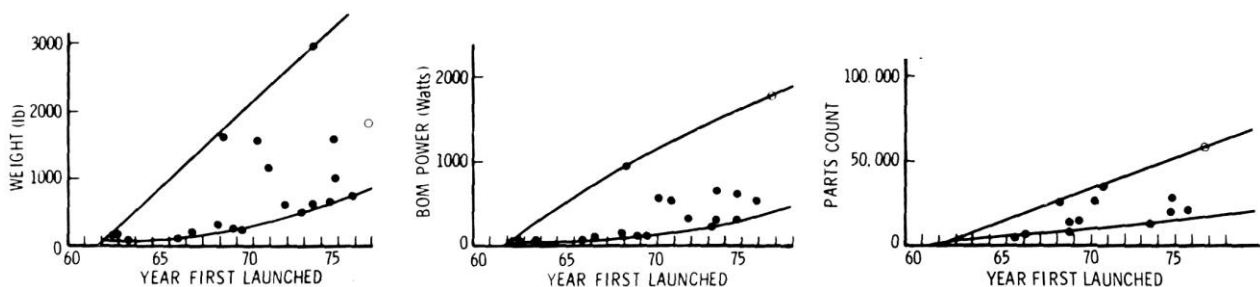


Figure 2-15: Growth of mass, power, and parts count of early communication satellites. Adapted from [85].

²⁹ The launch capability for Thor-Delta grew from roughly 45 kg in 1962 to over 900 kg in 1976 [85].

Levine reported a mean lifetime of communication satellites of about five years and an independency of lifetime and spacecraft size and complexity. He argued that more complex spacecraft simply require more reliability analyses and testing to guarantee a long lifetime. In one studied group, approximately 20% of experimental and military spacecraft failed during their first year on-orbit. This was not experienced in the subpopulation of communication satellites, and Levine attributed this to several NASA/military spacecraft being experimental and therefore simply not designed for a long lifetime. Finally, the study revealed that the first few flight models of a program often experienced more anomalies than later spacecraft of the same program, and that spacecraft that evolved from previous programs had a higher success rate in general. He concluded that the dominant factor in longevity of spacecraft is experience of the satellite developer [85]. In 1980 Baker & Baker [86] fitted a Weibull model into reliability data from 57 satellites. They found a decreasing hazard rate over time, best described by a Weibull model with a shape factor of $\beta = 0.87$ and a scale factor of $\theta = 700$ days. They concluded that space itself is not a harsh environment for spacecraft, since the failure rate of the studied group was decreasing and not increasing over time. This claim has of course to be seen in the light of the limited number of microelectronic components on those missions and the quite large structural sizes of those electronics [86].

Shockey [87] continued the work on longevity of space missions by a study on 104 orbital missions in 1981. He described the growth of projected and achieved lifetimes from the early sixties throughout the seventies. In the 1960s those values ranged from months to one year while in 1975 the median useful lifetime of a satellite used to be about 5.5 years. Also, he first described the problem of technological obsolescence, beginning in the 1970s, when more capable second-generation satellites were put into orbit. He further analyzed the distribution of on-orbit lifetime achieved in three time intervals: 1960–1964, 1965–1969 and 1970–1974. The first interval showed clear infant mortality, justified by Shockey with the infancy of the industry itself. Beginning in the second interval (1965–1969) wear-out was a factor to consider, and it manifested broadly between two and five years on-orbit. Shockey described this with the limited lifetime of early degradable components such as batteries, tape recorder and solar array drives. Within the last timeframe, early wear-out was solved for the most subsystems and the data appeared such as the constant failure rate region of the bathtub-curve. However, this appearance is not due to spacecraft failing purely out of random causes, as Shockey noted. He also described the learning curve of successive missions. In his group, the normal lifetime was often not achieved until the third spacecraft of a series. Subsequently he fitted the non-parametric reliability data of the 1970–1980 missions with a two-parameter Weibull function, with a shape factor $\beta = 1.9$ and a scale factor of $\theta = 5.2$ years. As we have seen before, this means that the dominant factor in this studied group was wear-out, not infant mortality (see Figure 2-16) [87].

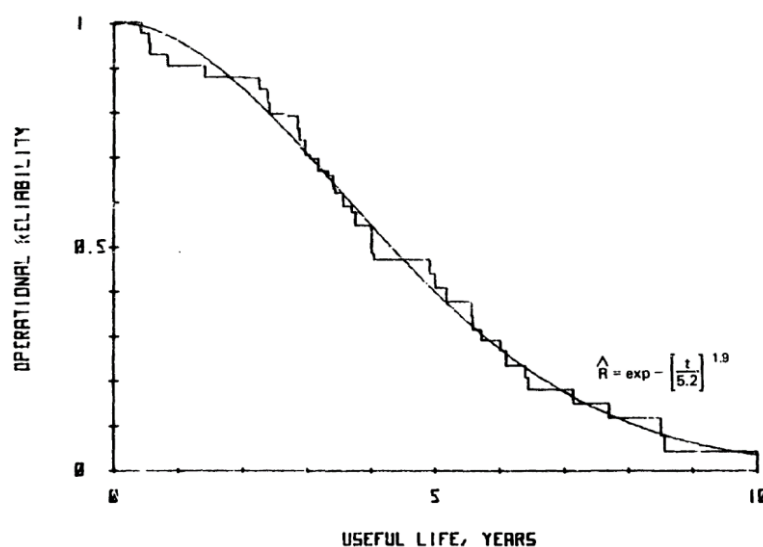


Figure 2-16: Parametric and non-parametric reliability of 1970–1980 spacecraft. Source: [87].

Concluding the study, Shockey compared his results with two other studies from the 1970's, conducted by the Research And Development (RAND) organization [88] and Thompson Ramo Wooldridge (TRW) [89]. These studies also used Weibull models, but their fits were dominated by infant mortality³⁰. In the group studied by Shockey this was not the case, although 41 spacecraft failed to survive their first year on-orbit. Interestingly, this was evenly distributed through the decade, and the failures were attributable to systematic causes rather than random ones. Finally, Shockey described that the reliability measures at GSFC are very well working for random defects, but emphasis should be put on reducing the probability of design and systematic errors [87].

In 1983, Bloomquist & Graham [90] reported on on-orbit anomalies and lifetimes of 44 NASA spacecraft operating from 1977–1982. Of the 606 anomalies found, the majority (44.1%) were of unknown cause, followed by design (14.9%), space environment (9.2%), and part failure (6.6%) as cause for anomaly [90]. Bloomquist [91] expanded his studies in 1984, describing the reliability of 374 spacecraft and 2,500 anomalies occurring on them. Looking back at his earlier work, he showed that fewer problem areas accounted for 50% of all anomalies in the post-1978 era than before that³¹. The five problem areas comprising 50% of all anomalies were in descending order: Scientific instruments, chemical propulsion, RF/EMI, telemetry sensing, and tape recorders.

The already in Subsection 2.1.1 mentioned 1985 study of Hecht & Hecht [54] for the Rome Air Development Center (RADC) was the most comprehensive study on the reliability of spacecraft up until then, specifically focusing on reliability prediction. 300 satellites launched between the early 1960s and January 1984 were used for it and many earlier studies, some of them described in this thesis, revisited. They confirmed the assumption of a decreasing failure rate for unmanned spacecraft. Although parts selection and quality control improved from the early days of spaceflight, they noticed a growth in design and environmental failures from pre-1977 compared to the 1977–1983 timespan. This growth was justified by the greater complexity of the satellites launched within the later timeframe. Out of the over 2,500 anomalies researched, 24.8% were due to design faults, 21.4% due to the environment and only 16.3% due to part errors. They concluded that the usage of constant failure rate models for reliability prediction is flawed since conventional part failures account only for a fraction of the overall numbers of failures and many other causes occur with a decreasing failure rate. Hecht & Hecht found a Weibull model with decreasing failure rate to fit best for their data. The Weibull function had a shape factor of $\beta = 0.28$ and a scale factor of $\theta = 255$ hours and clearly deviated from the exponential failure rate assumption (see Figure 2-17) [54].

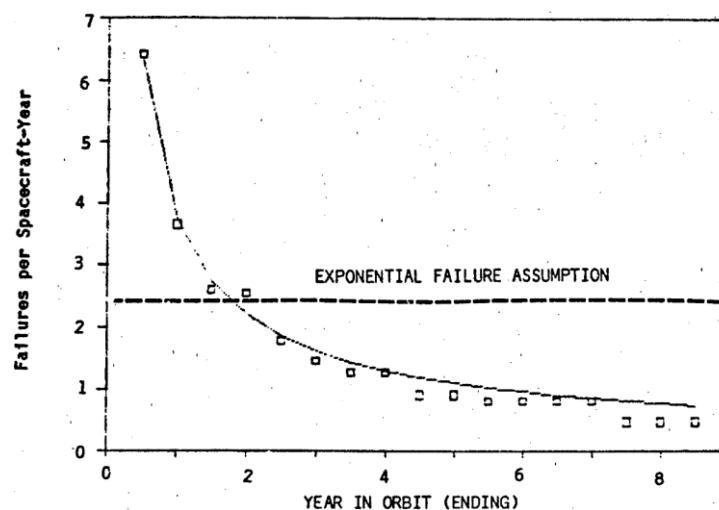


Figure 2-17: Weibull fit of on-orbit failure data of 300 satellites. Source: [54]

³⁰ TRW: $\beta = 0.911$, $\theta = 117$ days; RAND: $\beta = 0.859$, $\theta = 232$ days [87].

³¹ Pre-1978: nine problem areas; 1978 study update: seven problem areas; post 1978: five problem areas [90].

While the lifespan of satellites was growing over the decades, Ferguson & Davey [92] looked into the feasibility of orbital storage of satellite spares for upcoming constellations in 1985. They analyzed dormant and post-dormant properties of 155 electronic boxes on past missions, but reported that the number of satellite electronic equipment located at low earth orbits were too few to permit any statistical conclusions [92]. Hecht & Fiorentino [93] revisited the finding from Hecht & Hecht in a paper in 1988 and reported that albeit increasing complexity in spacecraft over the years, the fraction of critical failures per spacecraft went down from 10% pre-1977 to 3% after that. They explained this finding with much of the complexity increase in spacecraft being used for redundancy, thus shifting a portion of critical failures to a non-critical state.

In 1992 Ebeling [94] published his findings on reliability parameter estimation during the conceptual design of space vehicles. Based on previous work, he estimated a decreasing failure rate and a shape factor for subsystems of spacecraft operation in space environment to be $\beta = 0.311$ [94]. In the same year, Stevenson & Strauss [95] reported on the reliability of the Intelsat V satellite fleet. On 15 Intelsat satellites built between October 1976 and October 1988, a total number of 699 on-orbit anomalies was tracked. Stevenson & Strauss attributed the overall satisfying reliability of the satellites to the low incidence of systematic failure since the satellites researched originated from only two slightly different series of satellite types (Intelsat V and Intelsat V-A) [95]. Looking into 205 civilian geosynchronous satellites launched before 1993, Sperber [96] again confirmed generally decreasing on-orbit failure rates for unmanned spacecraft. In his paper he presented that the logarithmic number of satellite anomalies over logarithmic time always show a slope well below one³². Therefore, he concluded that the causes of the tracked anomalies are not random overstresses or wear-out phenomena, but errors in design or execution. Interestingly, he also reported an improvement factor of two on the failure rate between the first and the second spacecraft built. Overall, the probability of failure before end of life in his study group decreased by 30% per decade, even though the complexity of the spacecraft increased. This can be attributed to the increase in redundancy. Also of interest for that is that prior to 1989 no spacecraft had a planned design lifetime of more than 10 years [96].

Studying 132 orbiter and 9 interplanetary spacecraft, Krasich [97] also found shape factors below one most suitable for her data in 1996. Notably, the shape factors for failures on the studied interplanetary spacecraft were clustered around 0.5. She further presented the limited applicability of constant failure rate models, such as the MIL-HDBK-217F, for reliability prediction of spacecraft. Figure 2-18 shows the experienced failure rate for the Voyager spacecraft, a fitted Weibull model ($\beta = 0.43$, $\theta = 102,775$ hours) and the predicted reliability [97].

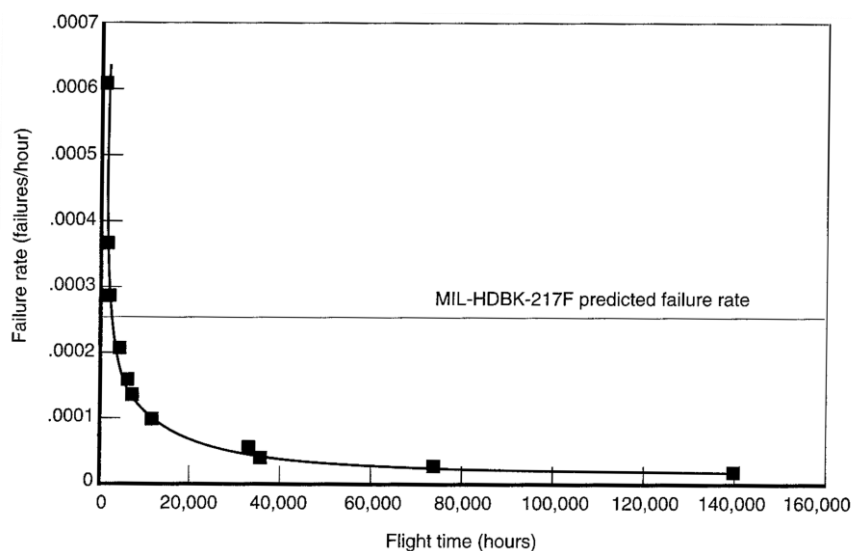


Figure 2-18: Predicted and experienced failure rate of the Voyager spacecraft. Source: [97], Adapted from [28].

³² A slope of one would be a constant failure rate. A slope below one is infant mortality.

In the same year, Hall & Blay [98] researched early orbit failures³³ in a study for ESA. They found 369 space vehicles that experienced a total of 900 early orbital anomalies and failures, with about 25% causing a loss of function. On the subsystem level, they found the Attitude and Orbit Control System (AOCS), mechanisms and communication to be the biggest areas of concern [98]. In 1996 NASA released the reference publication on spacecraft failures and anomalies attributed to the space environment, in which Bedingfield, Leach & Alexander reported on more than 100 cases of failures and anomalies on spacecraft caused by the space environment from 1974 to 1994 [50]. Later, Sarsfield [28] summarized the efforts of small satellite development since the early days of spaceflight in his book in 1998. He concluded that spacecraft reliability has been steadily improving since the beginning of spaceflight, and that if problems occur, they tend to be less significant. Also he noted that as the total number of failure decreases, design-related failures will play a more significant role [28]. In the same year Hecht [13] revisited his data from 1985 in a book chapter and concluded that design deficiencies and operational mistakes are still a major contributor to reliability problems of spacecraft. He recommended better distribution of design guidance, experience with prior systems and improved review techniques as ways to overcome that [13].

Although focusing on serviceable spacecraft failures from 1981 to 2000, Sullivan & Akin [99] also presented the rate of beginning of life (BOL) failures of spacecraft in their work. Considering the first 30 mission days as BOL, they found a 3% to 4% chance of BOL total failure and 3% to 4% chance of BOL partial failure for the studied group of spacecraft [99]. In 2003, Robertson & Stoneking [100] reported on on-orbit anomalies of 764 spacecraft launched between 1990 and 2001. 48% of all failures occurred in the first 10% of the mission design lifetime. They concluded that this indicates design flaws and latent manufacturing defects having a bigger impact on mission success than material contamination or fatigue, and recommended testing in a configuration as flight-like as possible [100]. Hoffmann, Green & Garrett [101] presented data on anomalies of long life outer planet missions in 2004. After studying the Voyager 1, Voyager 2 and the Galileo spacecraft, they concluded that redundancy prevented catastrophic failure in many cases for those missions. Overall, 3,300 incidents had been recorded in these three missions [101]. In 2005, Harland & Lorenz [102] gave a detailed review on satellite and rocket failures up to that point. Although little statistical data are given in their book, it can be highly recommended by the author of this work due to their level of detail in describing various launch and on-orbit failures and their root cause. In the same year, Maurer [36] also found the exponential distribution being too pessimistic for spacecraft reliability prediction. Studying seven robotic missions to Mars from 1990 to 2003, Green, Hoffman, Schow & Garrett [103] found nearly 1,400 anomaly reports and presented them in 2006. Although flight and ground software already took up 50% of all anomaly sources found (see Figure 2-19 left), Grottko et al. [67] noted that the number of software anomalies might be significantly underreported in the internal reporting system of JPL. Also interesting in the report of Green et al. [103] is the distribution of the subsequent corrective action taken after an anomaly was detected (see Figure 2-19 right).

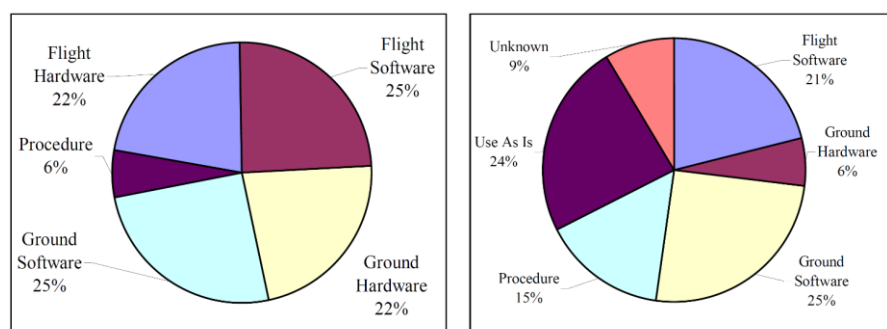


Figure 2-19: Anomaly source percentage (left) and subsequent anomaly corrective action (right) of seven Mars missions from 1990 to 2004. Adapted from: [103].

³³ For GEO satellites this would be the first 180 days on-orbit. Hall & Blay define early orbit as the period during which failures might occur before the mission is properly underway [98].

In one of the most comprehensive studies, Tafazoli [104] presented the on-orbit failures of 129 different spacecraft out of a group of more than 4,000 spacecraft launched between 1980 and 2005. In the analyzed group, the spacecraft mass and power capabilities were scattered over a wide range, thus mixing large satellites operating in geostationary orbit (GEO) with small satellites. Despite this, he observed that 41% of all failures happened within the first year on-orbit, many of them right after launch. He attributed this result to insufficient testing since electronic components on most of the satellites should have lasted at least 3-5 years of operation. He concludes his paper with three main areas of reliability improvement: adequate testing, redundancy, and flexibility (meaning the re-programmability for failure recovery) [104]. It should be noted that the latter two also invoke additional complexity in many cases, hence potentially decreasing system reliability again. In 2008, Rodiek & Bradhorst [105] also found infant mortality, thus a decreasing failure rate, best fitting for their study on solar array reliability in satellite operations. In 2009, Ogamba [106] described the on-orbit reliability data of nine spacecraft of the Defense Meteorological Satellite Program (DMSP). 65% of all failures tracked were in the payload subsystem of these satellites and the biggest root cause for those failures was workmanship (40%). Ogamba subsequently analyses the predicted and measured failures per million hours for the thirty-six data recorders (four per satellite) of the fleet. Using Bayesian analysis, a posterior failure rate of 5.7 failures per million hours, less than the predicted rate of 6.318 failures per million hours, was found. Notably, two separate failure mechanisms were registered in the group of tape recorders, one clustered around 16,000 hours (five failures), the other one around 50,000 hours (five failures). Though this was modeled with one Weibull function ($\beta = 1.99$), Ogamba reported that further investigation revealed that the recorders seemed to be from two distinct batches [106], which could explain the different failure behavior. In 2011, both Hecht & Hecht [21] and Hurley & Purdy [107] published book chapters on reliability of spacecraft. As before, Hecht & Hecht associated most failures in spaceflight to poor workmanship and design [21]. Hurley & Purdy summarized earlier findings and concluded that since most failures occur early in the life of spacecraft, the constant failure rate approach is not suitable for spacecraft reliability prediction, yet still often used. They further noted that many reports corroborated the assumption that the accumulation of failures within the first year of a mission is driven by design or workmanship faults that simply expose themselves early in a mission [107].

In 2012 Monas, Guo & Gill [108] analyzed 296 orbit anomalies of 222 different small satellites launched between 1990 and 2010. For their analysis they classified the group further into Picosatellites (< 1 kg), Nanosatellites (between 1 kg and 10 kg), Microsatellites (between 10 kg and 100 kg) and Minisatellites (between 100 kg and 500 kg)³⁴. All groups and the overall group of small satellites were studied with nonparametric and parametric functions, and infant mortality was found as a clear pattern in all small satellites. The parametric, two-parameter Weibull function for small satellite reliability had a shape factor of $\beta = 0.3134$ and a scale factor of $\theta = 3,062$ days. For the different mass-classes they found: Picosatellites ($\beta = 0.3476$ and $\theta = 60$ days), Nanosatellites ($\beta = 0.4538$ and $\theta = 277$ days), Microsatellites ($\beta = 0.2928$ and $\theta = 10,065$ days) and Minisatellites ($\beta = 0.3938$ and $\theta = 4,400$ days). The similar shape factor of all mass groups is explained by them with an inaccuracy introduced by the Maximum Likelihood Estimation (MLE) method because of a lack of observed failures in the Pico- and Nanosatellite group [108]. Unfortunately, no overall statistic on the CubeSat class of spacecraft and no further details on non-parametric data or underlying number of satellites or failures are given in their paper. The group continued their research and published two updates in 2014. In their first update, utilizing the same data as in 2012, Guo, Monas & Gill [109] published a paper on updated parametric models of small satellite reliability. Using a Markov Chain Monte Carlo (MCMC) approach, they found shape and scale factor to be very close³⁵ to the MLE estimation provided in their 2012 paper. Goodness-of-fit is reported to be 0.9233 for MLE and 0.9318 for MCMC. They further analyzed the parametric and non-parametric reliability of subsystems of small satellites, and later classified the data in the same sub-groups as in 2012. Again, the MCMC approach showed little differences

³⁴ We will discuss this classification in Section 2.3.

³⁵ Shape factor of $\beta = 0.3136$ and a scale factor of $\theta = 2,723$ days.

to the MLE approach. In the 2014 paper they provided the number of failures for the mass groups, reporting 17 failures (14 observed) in the Picosatellite class and 19 failures (13 observed) in the Nanosatellite class [109]. However, as can be seen in Figure 2-20, the overall number of satellites studied for both mass classes is limited, so, for instance, a failure of one Picosatellite at approximately $t = 0.1$ years would have big impacts on the non-parametric function. The same holds true for the failure of one Nanosatellite at approximately $t = 1$ year. Also, CubeSats are again not treated as an own class of satellites but split into two different mass categories and probably mixed with other not CubeSat-standardized satellites.

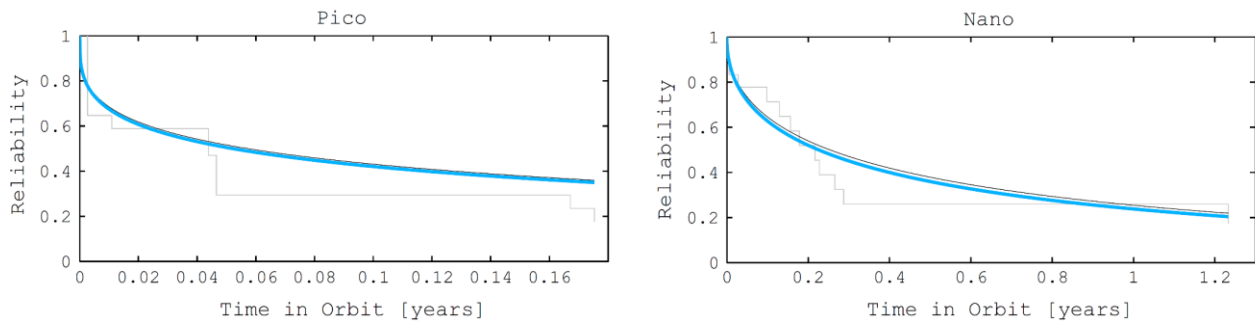


Figure 2-20: Non-parametric and parametric reliability of Pico- (left) and Nanosatellites (right). Source: [109]

In their second paper in 2014 Guo, Kolmas & Gill [110] specifically studied the reliability of spacecraft below 50 kg. For that purpose, they built a database containing 141 anomalies on 117 satellites, and more important for this work, 44 failures on 30 successfully launched CubeSats. Again, unfortunately not classifying their data in an overall CubeSat group, they subdivided their results into parametric models of different sizes of CubeSats, namely 1U, 2U and 3U³⁶. The parametric models were: 1U CubeSat ($\beta = 0.3697$ and $\theta = 136.6$ days), 2U CubeSat ($\beta = 0.6558$ and $\theta = 79.4$ days) and 3U CubeSat ($\beta = 0.4789$ and $\theta = 246.1$ days) [110]. Although the 1U and 3U envelope have similar shape and scale factors, it can be noted that the overall limited number of CubeSats in this database and the subsequent split into different sizes of CubeSats, without studying the overall CubeSat class, makes it statistically difficult to draw conclusions for this class of satellites. The slightly higher factors for the 2U class can be explained by the limited number of failure cases (two) used in their work.

2012–2016 saw several more reports on spacecraft reliability: Gorbenko, Kharchenko, Tarasyuk & Zasukha [111] studied rocket and spacecraft failures in the 2000–2009 timeframe, and Fox, Salazar, Habib-Agahi & Dubos [112] presented a satellite mortality study based on 722 unique spacecraft in 2013. Fox et al. found a two-parameter Weibull function with $\beta = 0.65$ and $\theta = 166$ years to be suited best for the group of studied spacecraft and compared this to a classical exponential model ($\lambda = 0.01667$ per year) [112]. In 2014 the RAND Corporation [51] reported on the benefits of a centralized anomaly database, in which all root causes and investigation results could be shared. Pelton concluded in his 2016 book chapter that 50 years of experience shows that once a satellite is deployed and tested out (i.e., past the infant mortality zone), it has a 90% chance of achieving its projected lifetime [19]. The need to reduce infant mortality cases by design and testing efforts is clearly shown by this, considering that the timeframe of testing out can range between several days (CubeSats) to half a year (GEO satellite).

In 2016, Palla, Peroni and Kingston [113] studied a sample of 798 spacecraft with a mass below 1,000 kg, launched between January 2000 and December 2014. Similar to Guo et al., they grouped the satellites in mass categories and found a single-parameter Weibull fit of $\beta = 0.3456$ and $\theta = 27,441,000$ days for the 1-10 kg spacecraft class [113]. In the view of the author of this thesis, these results should be taken with care, since Palla et al. only found 24 failures in the studied group of 281 spacecraft in the 1-10kg range. In

³⁶ As we have seen before, CubeSats are standardized satellites. As 1U means approximately a 10 x 10 x 13 cm satellite, 2U doubles (10 x 10 x 26 cm) and 3U triples (10 x 10 x 40 cm) this volume.

a second publication in 2016, Peroni et al. [114] continued the work, using a sample of 1,086 LEO spacecraft, launched between January 2000 and December 2014. A single-parameter Weibull fit of $\beta = 0.470$ and $\theta = 322,740$ days is obtained in this paper, while the reliability of the 1-10kg class of satellites is reported similarly to their other publication.

We will conclude the subsection on historical findings on reliability of satellites by summarizing the work of the research group of Saleh and work that is based on their findings. In their first publication in 2009, Castet & Saleh [115] presented a non-parametric and a parametric statistical analysis on satellite reliability, based on a study group of 1,584 satellites launched between January 1990 and October 2008. Arguing with the usage of shape factors bigger than one in past studies (thus using increasing failure rates), they motivated their work with needed input for satellite developers for satellite test & screening programs, redundancies, and reliability growth plans. In the studied group, they found 98 failures and 1,486 censored times for the 1,584 satellites. Censoring occurs when the satellite is retired or still operational at the end of the observation window (October 2008). After presenting the non-parametric estimation, Castet et al. linear last square fitted a single-parameter Weibull function with a shape factor of $\beta = 0.3875$ and a scale factor of $\theta = 8,316$ years (see Figure 2-21). They later presented this work condensed in a journal paper in 2009 [116].

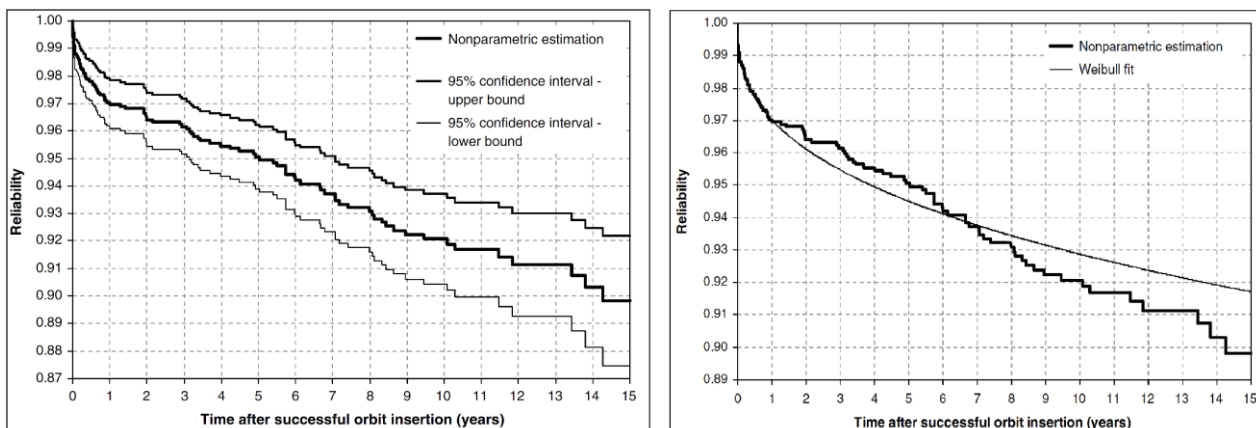


Figure 2-21: Non-parametric (left) and parametric (right) reliability of the satellites. Source: [116]

In both papers they continued by presenting the allocation of failures to the different subsystems of satellites. They showed that solar array deployment at $t = 0$ days and telemetry, tracking, and command (TTC) as well as thruster/fuel problems are the most affected subsystems within the first years. Later, around year 10, gyro failures become the predominant source and very late (year 14) battery failures also become increasingly important. They concluded both papers by stating that their studied group of satellites show clear infant mortality and that current shape factors used by the industry are not correct [115].

In a third paper in 2009, Castet and Saleh [117] presented the first MLE estimated single-parameter Weibull function of their studied satellite group. With a shape factor of $\beta = 0.4521$ a scale factor of $\theta = 2,607$ years, the resulting Weibull function of the overall satellite reliability showed some deviations from the linear last square fitted one. They subsequently fitted their subsystem reliability data also with non-parametric and parametric models, and simulated the satellite reliability by a Monte Carlo chain simulation of all subsystems. The resulting curve followed the non-parametric estimation within one percent point over studied lifetime of 15 years [117]. In the same year, Dubos, Castet and Saleh [15] analyzed if the spacecraft size is a factor on the reliability of satellites. They used a reduced sample (1,444 satellites) of the earlier group with 415 small satellites (below 500 kg), 554 medium satellites³⁷ (between 500 kg and 2500 kg), and 475 large satellites (over 2,500 kg) [15]. This supports recommendations of earlier studies to only use reliability data of similar classes of satellites for reliability studies. The non-parametric and parametric

³⁷ In this sample satellites belonging to the IRIDIUM constellation were excluded.

analysis revealed clear infant mortality in the small class of satellites and a monotonous increase of the shape factor of the fitted two-parameter Weibull distribution with satellite's mass. The parametric models were: small satellites ($\beta = 0.3224$ and $\theta = 21,414.5$ years³⁸), medium satellites ($\beta = 0.5973$ and $\theta = 1,469.2$ years) and large satellites ($\beta = 0.6794$ and $\theta = 291.4$ years). Thus, all satellite classes experienced a decreasing failure rate over time. Within the non-parametric fits, a wear-out behavior, more distinct for large satellites than for smaller satellites, was described by Dubos et al. after approximately 6.5 years (see Figure 2-22). To capture that, they continued by fitting a 2-Weibull mixture model for each mass category, depicted in Figure 2-22 for the small category. These fits and their coefficients will be discussed in more detail in Section 4.1.

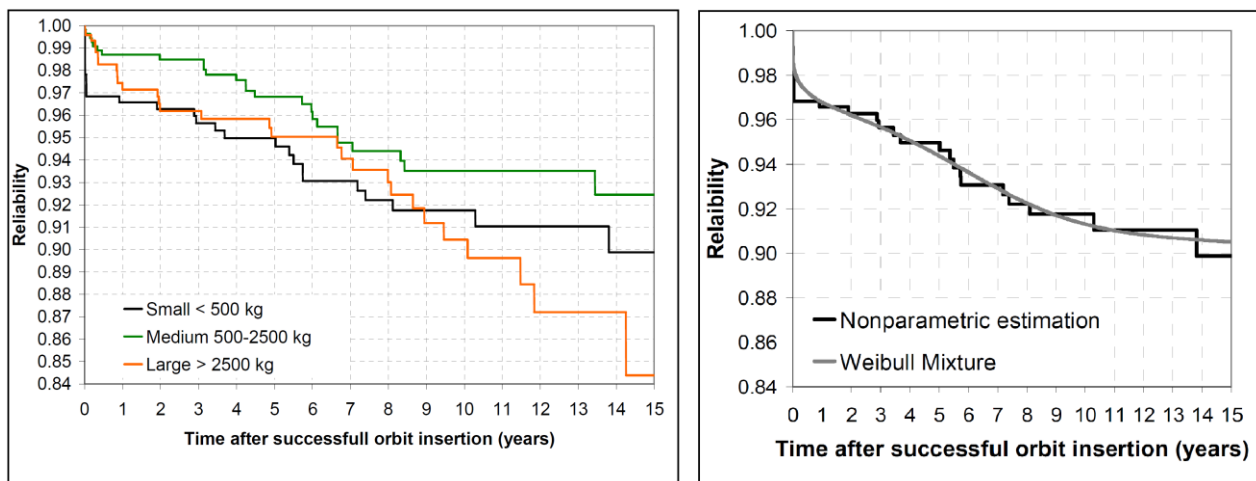


Figure 2-22: Non-parametric reliability (left) of different mass classes of spacecraft and parametric 2-Weibull function fit (right) of small satellite reliability. Source: [15]

In their paper, Dubos et al. later used conditional reliabilities to show that the reliability of small and large satellites overlaps significantly between $t = 0.5$ years and $t = 8$ years. They concluded that small satellites, while experiencing roughly the same decrease in reliability over time as large ones on later stages, exhibit more extensive infant mortality in their earlier life. They further hypothesized that less stringent testing, heavy reliance on COTS components, less redundancy, and less shielding used for small satellites could be reasons for that [15]. As we have seen before, design and workmanship errors, not detected due to limited test resources, could be added to that list.

The same group of researchers slightly updated their findings in a 2010 journal paper [118] on the reliability of different mass categories of satellites. In their paper, they further reduced their sample size to 1,394 satellites, thus eliminating 50 satellites from the 2009 paper's sample. Also, in difference to their first paper on this topic, they censored satellites that failed after they reached their lifetime [118]. The last assumption particularly influences the small satellite category, in which many satellites are traditionally operated well beyond their (usually relatively short) design lifetime. We will revisit the effects of this assumption in Section 4.1. With the adapted groups, they again fitted non-parametric and parametric models³⁹ and despite these differences, overall reached the same conclusions in their paper as in [15]. In their last paper in 2009, Hiriart, Castet, Lafleur & Saleh [119] compared the reliability of LEO, medium earth orbit (MEO), and GEO satellites. Based on the restriction to 1,488 satellites of the aforementioned group, they found 70 failures in 882 LEO, two failures in 111 MEO, and 22 failures in 495 GEO satellites. Figure 2-23 depicts the non-parametric reliability of the LEO and GEO sample.

³⁸ We will discuss the implications of such large scale factors in chapter 4.1.

³⁹ The coefficients of the two-parameter Weibull models were: small satellites ($\beta = 0.2519$ and $\theta = 893,150.6$ years), medium satellites ($\beta = 0.4492$ and $\theta = 18,215.6$ years) and large satellites ($\beta = 0.6926$ and $\theta = 273$ years).

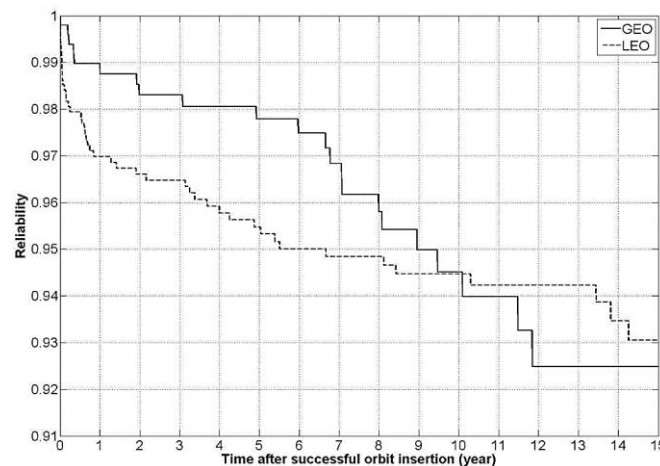


Figure 2-23: Non-parametric satellite reliability of LEO and GEO satellites. Source: [119]

Both the non-parametric and the parametric models showed a decreasing failure rate for the LEO and GEO class of satellites. For MEO, having their overall small numbers of failures in mind, an increasing failure rate was found⁴⁰. They also used 2-Weibull mixture functions to model the parametric reliability, concluding that differences in space environment, power cycles, and overall budget constraints (leading to less testing, redundancy and restricted part selection) might cause the differences between LEO and GEO spacecraft. Using their original group of 1,584 satellites, Castet & Saleh [120] later applied a 2-Weibull mixture function to their nonparametric data to obtain a better fit. The function, composed of a mixture of two Weibull functions with $\alpha_1 = 0.9484$, $\beta_1 = 0.2575$, $\theta_1 = 982,100$ years and $\alpha_2 = 0.0516$, $\beta_2 = 1.997$, $\theta_2 = 10.2$ years, follows the non-parametric data with better accuracy than the original one. Figure 2-24 shows the 2-Weibull mixture function. According to Castet & Saleh, the function with $\beta_1 = 0.2575$ captures infant mortality while the function with $\beta_2 = 1.997$ allows the wear-out portion of the overall reliability to be considered. This is arguable since the scale factors of both functions cause an opposite effect. Furthermore, they hypothesized that higher order mixture distributions are superfluous since their 2-Weibull mixture distribution showed a quasi-random dispersion of the residuals [120]. We will further discuss both assumptions in Section 4.1.

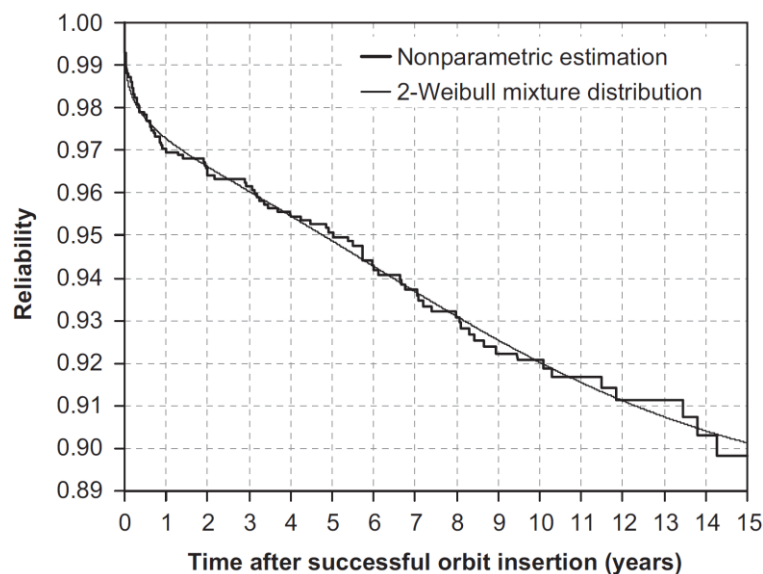


Figure 2-24: 2-Weibull mixture fit of satellite reliability. Source: [120]

⁴⁰ The coefficients of the two-parameter Weibull models were: LEO satellites ($\beta = 0.3473$ and $\theta = 34.04896$ years), MEO satellites ($\beta = 1.6347$ and $\theta = 79.4$ years) and GEO satellites ($\beta = 0.7190$ and $\theta = 582.5$ years)

Castet & Saleh summarized their papers in a 2010 book chapter [121] and further researched multi-state failures (i.e., failures that don't lead to a loss of the spacecraft but to degradation) of satellites in the same year [122]. Many findings of both authors were put in a book in 2011 [22], and some parameters for the 2-Weibull mixture models were updated. These updated models (and the former models) are depicted in Section 4.1. In 2012 Castet & Saleh continued with work on the survivability for spacecraft and space-based networks, using stochastic Petri nets (SPNs) for modelling purposes [123]. Also in 2012 Peng & Zhang [124] used a modified Weibull extension distribution with a bathtub-shaped failure rate function on the original data of Castet & Saleh. The modified function, based on work by Xie, Tang & Goh [125] defines the satellite reliability function as:

$$R(t) = \exp \left\{ \lambda \cdot \alpha \cdot \left\{ 1 - \exp \left[\left(\frac{t}{\alpha} \right)^\beta \right] \right\} \right\} \quad (12)$$

The parameters obtained by Peng et al. were $\alpha = 0.196$, $\beta = 0.221$ and $\lambda = 0.042$. The resulting fit was within 0.25% of average error with regard to the non-parametric estimation by Castet & Saleh. Furthermore, Wayer, Castet & Saleh [126] specifically researched spacecraft attitude control subsystems in 2013, using the same sample of 1,584 satellites. In their newest publication from 2017, Saleh, Geng, Ku and Walker II [127] used a new sample of 162 electric propulsion equipped satellites launched between January 1997 and December 2015. Since the reliability of specific subsystems is only of minor interest for this work, we won't go into more detail on these studies.

Summarizing this subsection, as Cheng & Smith [128] pointed out, satellites rarely fail due to defective parts or environmental factors. Most failures are caused by engineering errors and are often too subtle to be found during routine review and verification [128]. As presented in the paper of Lowry [59], high risk technologies have two risk dimensions: interactions and coupling. Risky in interactions means to have unfamiliar or unexpected sequences, and that complex interactions are not immediately comprehensible. Coupling means to have systems of multiple time-dependent processes that cannot be delayed or extended [59]. Spacecraft combine both, tight coupling and complex interactions, and are therefore per se risky systems. This combination is especially susceptible to human errors, thus engineering faults, as systems with tight coupling and complex interactions are at least difficult and sometimes impossible to fully comprehend. Engineering faults, as we have seen from the various statistics, will emerge shortly after launch, leading to an overall decreasing failure rate over a variety of satellite classes and orbits.

Due to limited resources in testing and redundancy, small satellites, especially CubeSats, are vulnerable to engineering mistakes and therefore experience a higher infant mortality than bigger satellites, despite their reduced complexity. The time dependent reliability data on CubeSats, necessary for identifying failure patterns, is limited and more work is needed on parametric modelling of different mass classes of satellites. As pointed out by Reeves [42], it is desirable that the satellite manufacturer should debug the system thoroughly prior to its use in the field and therefore limit the occurrence of infant mortality. Wear-out, emerging in some bigger missions, must be prevented too, but infant mortality seems to be the best target to reduce failures in current small satellite and CubeSat missions. Also, we have seen that random part errors hardly cause satellite failures, opposing many of the current system reliability prediction models used in satellite engineering. Sarsfield [28] reported an 11% rate of random part errors, and that is substantiated by other research shown in this subsection. Lifetime expectancies of satellites grew from half a year or a year to 15 years or longer, and today's testing comprises 20-35% of the overall spacecraft manufacturing costs [19]. Though the overall number of spacecraft failures is decreasing, and failures are less severe than before [28], the prevalence of infant mortality, the substantial contribution of engineering failures and the limited resources (time, manpower, money) of small satellites and CubeSats make a more thorough study of this topic necessary. In the next section, we will briefly look into reliability prediction and reliability assessment in traditional space missions.

2.2 Reliability Prediction, Assessment and Assurance in Space Missions

In this section, we will summarize important approaches used in traditional spaceflight applications when dealing with reliability, namely reliability prediction, reliability assessment and reliability assurance. First, a distinction between these different terms is necessary. Reliability prediction means to calculate a system or subsystem reliability, based on the structure of the system or subsystem and the failure rate of its parts. Reliability assessment always implies getting data out of reliability tests or from in-the-field and statistically evaluating that [39]. It is important to understand that data from reliability assessments can also be used as input for reliability prediction at later stages. For example, on-orbit data of the first of a series of satellites or part reliability out of accelerated tests in simulated space environment could be used for that. Reliability assurance is defined much broader and can include both, prediction and assessment strategies. In general, a wide range of different practices is used in today's traditional space missions to assure reliability, from redundancy over testing to mission simulation and training [107]. Reliability assurance overlaps with the efforts of mission assurance, while the latter term is used more often. In the following subsections, we will first deal with reliability prediction since it is traditionally used firstly, in order to prove that requirements are met by certain designs. Different tools are available for that, ranging from the already mentioned handbook-based approaches to physics of failure techniques. Generally, reliability improvement can result from both, reliability prediction as well as reliability assessment. In the former case, calculations show the necessity to change designs or individual parts based on a calculated failure rate or the consequences of certain failures. In the latter case, it can mean that a test showed flaws in a design or a part is not suitable for space environment and has to be replaced. We will not specifically focus on reliability improvement and how to implement these changes, but rather see them as logical step after errors are detected. Traditionally, reliability improvement was either done by simplification of the design, by selecting components with a lower failure rate or by adding additional redundancy [13]. As we have seen in the last section, this might not be feasible or the full set of solutions possible for modern, complex space systems as well as small satellites. Software failures and engineering failures are aforementioned examples of failures, in which traditional reliability prediction and improvement will not help. After dealing with reliability prediction, we will focus on reliability assessment and summarize a few state-of-the-art testing methods used in space projects. In the last section, we will cover different aspects of reliability assurance. Since all three areas cover a wide range of different tools and methods and many of them are topics of ongoing research, only a selected overview can be given.

For all topics discussed it is important to consider certain aspects of reliability in space missions. Historically, the harsh environment of space and the remoteness of spacecraft from earth demanded that satellites are designed to the highest reliabilities possible, incorporating many redundancies and testing the systems under extreme conditions here on earth [19]. Small satellites were always an exemption to this rule, simply because their limited envelope and project resources restricted both the reliability design goals and the applicable methods. Furthermore, it is important to understand that reliability is always linked to a certain timeframe, and given in percentages of success rather than just fail/pass. High reliability is not equal with spacecraft longevity, and a spacecraft designed to operate for one year can fail on the 366th day and be considered 100% reliable [28]. That is important to consider, especially for small satellite missions, since many of the traditional missions focus on 100% space mission success and mandate for zero defects on every level of the system, accepting the often associated cost schedule growth [129]. For GEO satellites, a maximum outage of about 100 minutes per year and 99.98% reliability is common [19]. Thus, the zero-defect approach might be the right choice. However, for small satellites and especially CubeSats, in which fast demonstration of technology is a major focus, it is clearly not. As mentioned before, in both cases delays can be perceived as the spacecraft achieving zero percent of reliability, and both, cost and schedule growth increase the chance of project termination, also leading to zero percent reliability. Saleh and Marais [130] showed that current value calculations for spacecraft often falsely assume that the spacecraft stays 100% reliable throughout its life, overestimating the system's value and leading to flawed investment

decisions. All of this is different in the world of human spaceflight, in which delays, and cost growth can and should be accepted. Grumman, while embracing a zero-fault approach for the Lunar Excursion Module, produced 29 test articles of the module to achieve this goal [19].

Small satellites and CubeSats can benefit from the advancements of reliability assurance on terrestrial systems, such as the Six-sigma approach. Instead of traditionally custom-fitting specialized parts to unique subsystems, CubeSats and small satellites can utilize high-volume, high quality COTS-parts, and also buy complete, space-proven subsystems off-the-shelf. We will further discuss this in Subsection 2.2.3. In general, space systems tend to be in the upswept right tail of the curve depicted in [28]. Thus, the high reliability levels are bought with high total cost. For small satellites and CubeSats resources are often very limited but approaches from terrestrial product development might help to alter the curve.

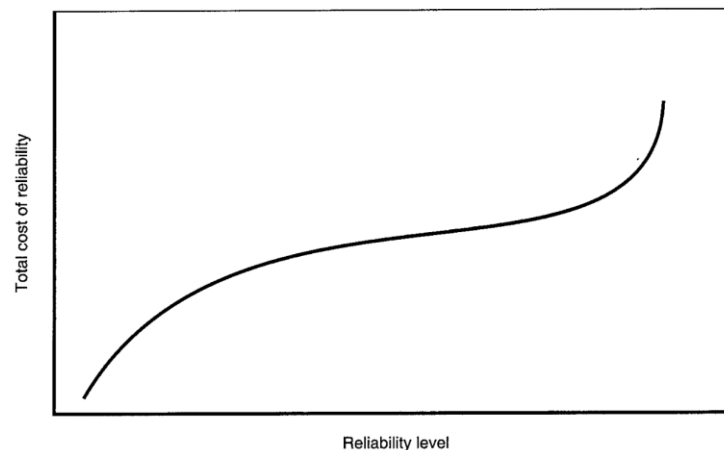


Figure 2-25: The General Shape of the Cost-Reliability Curve. Source: [28]

2.2.1 Reliability Prediction in Space Missions

As we have seen in Subsection 2.1.3, the definition of reliability itself evolved over time into as what it is perceived today. Birolini [39] noted that until the 1960s a reliability target of a product was deemed to be reached when the item was found to be free of failures at the time it left the producer. Nowadays, customer expect a product to be failure free not only at the time of purchase, but also for a stated time interval. To predict the probability of that, reliability prediction modeling was established in the 1950s by the Advisory Group for the Reliability of Electronic Equipment (AGREE). Subsequently, the first reliability prediction handbook of electronic equipment, the already mentioned MIL-HDBK-217F was published by the US Navy in 1962 [131]. Today, reliability prediction serves in many ways during the whole product lifecycle. It can help to achieve a reliable design, support the manufacturing process, and be used to compare different designs by quantifying the expected in-the-field reliability. Later in the product lifecycle, reliability prediction can help to identify reliability problems, to predict warranty-cost, and to assess warranty risks [131]. The work of Johansson [132] gives a broader overview over the variety of methods used. She grouped the methods into methods for fault avoidance, such as part count and part stress analysis, and into methods for dependability analysis, either bottom-up (e.g., FMEA) or top-down (e.g., FTA). Another classification, reported by Goel & Graves [40], is distinguishing between empirical-based models, based on past experience, and physics-of-failure models. We will use no classification in this work but rather go step by step through the different methods used in today's spaceflight applications. While doing so, it is also important to understand the assumptions used in many of the underlying models, and the transition of failure rates generated by reliability prediction or reliability assessments into dependability assessments such as FMEA.

As heard before, the majority of reliability prediction methods assumes that the design of a system is perfect and all stresses are known, so that only random failures occur [44]. Furthermore, most of the models assume that systems fail as a result of failures of parts, and those parts fail predominately as a result of exposure to stress [133]. As we have seen from in-the-field data in Subsection 2.1.3, this is not the case. Also, inaccuracies in the data used for these component failure rates, simplifications in the mathematical modelling, and lacking consideration of internal or external interference are points of criticism for reliability prediction [39]. Although research has already been carried out on applying non-constant failure rate models to systems, and the results agree very good with field data [133], constant failure rate models are still very common in spaceflight applications, despite other evidence [54], [134]. Prediction methods can be applied to economically predict reliabilities of different designs and to support the decision process in early phases. But as Hurley Jr. & Purdy [107] noted, predicting on-orbit success or failure of the mission, or decisions such as implementing full redundancy as a hard requirement or mandating for Class S electronic parts on all levels cannot be based solely on reliability prediction models. This misuse often leads to major reliability decreases such as program cancellation due to cost or schedule overrun, leading to a reliability of zero. Nevertheless, since system tests can only be performed late in the development, reliability prediction, if carefully applied, can help engineers on critical decisions early in the product design process [131].

We will start with **part count** and **part stress** methods, which are both empirical-based models, used for failure avoidance early in the design process. Part count is the simplest approach and can be used when a rough estimate of reliability is required and the analyzed system either has no redundancies or its redundancies can be neglected. Thus, the sum of all failure rates of the system equals an upper bound of probability of system failure [39], [132]. Empirical data are summarized in handbooks such as the MIL-HDBK-217F [24], in which the failure rate of the system obtained by the part count method is defined as:

$$\lambda_{Equip} = \sum_{i=1}^{i=k} K_i \cdot (\lambda_g \cdot \pi_Q)_i \quad (13)$$

with λ_{Equip} being the total equipment failure rate in failure per 10^6 hours, k the number of different generic part categories in the equipment, K_i the quantity of the i^{th} generic part, λ_g the generic failure rate for the i^{th} generic part in failure per 10^6 hours, and π_Q the quality factor for the i^{th} part [24]. Handbooks give empirical values for λ_g and π_Q . The reliability of the connections between the parts (interfaces) is not considered.

The part stress method, also used for early design decisions, takes more factors into account than the part count method. Also based on empirical data provided by handbooks and the negligence of redundancies, the upper bound of the failure rate of a system is again calculated by the sum of the part reliabilities. For the part failure rate, an extended model of the part count method is used, as seen for example in the MIL-HDBK-217F [24]:

$$\lambda_p = \lambda_b \cdot \pi_T \cdot \pi_A \cdot \pi_R \cdot \pi_S \cdot \pi_C \cdot \pi_Q \cdot \pi_E \quad (14)$$

where λ_p is the part failure rate, and λ_b the base failure rate. Depending on the specific electronic part, this base failure rate gets modified by factors π_T (temperature factor), π_A (application factor), π_R (power rating factor), π_S (electrical stress factor), π_C (contact construction factor), π_Q (quality factor), and π_E (environment factor) and potential additional factors not mentioned in the equation (e.g., package type factor, die complexity factor). Different from the part count method, the reliability of the connections between the parts can be considered in the part stress method. Both methods rely heavily on the up-to-dateness of the used handbooks.

The **physics-of-failure** approach assumes that failure mechanisms are governed by fundamental mechanical, electrical, thermal, and chemical processes [40], with the objective of finding the root cause mechanism of part or system failure. A good overview of the method is given by McLeish & Tomczykowski in their paper from 2013 [135]. The method emerged in the 1960s and was developed to overcome limitations of classical, empirical based models by studying all aspects of failures, and tracing them back to their root causes. In particular, the perception that failures are random and unavoidable can be overcome by the physics-of-failure approach since corrective actions can be applied once the root cause is known. Early examples of the physics-of-failure approach are Finite Element Analysis for mechanical stress issues and similar models for thermal stress [135]. As Gericke [136] and Goel & Graves [40] noted, this approach is particularly useful for wear-out mechanisms on the discrete part level. As for the disadvantages, it cannot be used to estimate the field reliability, and it is a highly complex process, which requires deep knowledge on materials, processes, and failure mechanisms and thus is not practical for assessing entire systems [40]. Furthermore, engineering errors or process defects are not covered by this method. Finally, although started as a replacement of empirical models, some handbook-based methods include the physics-of-failure approach in their models.

Before giving a brief overview of reliability prediction methods for dependability assessments, we will look on the widely used **handbooks for reliability prediction**. Starting in 1962 with the MIL-HDBK-217F, a variety of handbook-based prediction methods has emerged, as summarized in Table 2-5. As already pointed out, all handbook-based prediction methods share the fundamental risk of being outdated (e.g., having outdated empirical data and not keeping pace with the shrinking scale sizes of electronics).

Table 2-5: Overview of handbook-based reliability prediction methods. Source: [137]

Procedural method	Last updated year	Country of origin	Status
MIL-HDBK-217	1995	USA	Active
GJB/Z 299	2006	China	Active
Telcordia SR-332	2016	USA	Active
PRISM	2000	USA	Active
RDF-2000	2000	France	Active
217Plus	2015	USA	Active
FIDES	2009	France	Active
Siemens SN29500	2013	Germany	Active
NTT Procedure	1985	Japan	Canceled
British Telecom	1994	UK	Canceled
HRD-5			

This is especially a problem for the widely used **MIL-HDBK-217F**, which saw its last update in 1995 [25], when the typical feature size of electronics was around 500 nm, while the 10 nm feature size is state-of-the-art in today's consumer electronics. Furthermore, some active and passive components have not even been invented when the handbook was last updated, for example niobium capacitors and insulated gate bipolar transistors [138]. In general, as mentioned before, the MIL-HDBK-217F only assumes random part failures, thus a constant failure rate on part level. Implicitly this also means that every failure can be traced back to a part, and there is no failure related to design or quality-related problems [40], [54], [93]. Though the handbook allows both, part stress and part count as well as 14 separate environments for operations to be chosen [136], some values provided in the handbook have unknown origin, as Maurer [36] reported. The handbook itself states that, "*Hence, a reliability prediction should never be assumed to represent the expected field reliability as measured by the user (i.e., Mean-Time-Between-Maintenance, Mean-Time-Between-Removals, etc.). This does not negate its value as a reliability engineering tool; note that none of the applications discussed above requires the predicted reliability to match the field measurement.*"⁴¹

⁴¹ Department of Defense, "Military Handbook: Reliability Prediction of Electronic Equipment," MIL-HDBK-217F, Dec. 1991, pp.3-2.

To summarize, although it is no longer maintained by the Department of Defense and not intended to provide realistic field reliability predictions, the MIL-HDBK-217F is still widely used in the space industry as a basis for management decisions, for which it was never intended to. We will look at examples and some field-returned data at the end of this subsection. Since the last update of the MIL-HDBK-217F, several other handbook-based reliability prediction methods emerged (see Table 2-5). To describe all of them would exceed the scope of this work, so we will focus on three widely used methods in the next paragraph: FIDES, PRISM, and IEC 62380.

FIDES was developed by the French Ministry of Defense for aerospace, defense and civil purposes and intended to replace the MIL-HDBK-217F and **IEC 62380** due to their obsolescence. Similar to other handbook-based approaches, FIDES is used to calculate the failure rate of an electronic system by the sum of its parts, thus not taking redundancies into account. As opposed to the MIL-HDBK-217F, it includes a physics-of-failure approach and is updated regularly [139]. Feedback from orbit shows that FIDES provides a more realistic prediction of the reliability of spacecraft as we can see at the end of this subsection. Nevertheless, similar to MIL-HDBK-217F, FIDES is limited with respect to a broad variety of common failures occurring in spacecraft stemming from engineering flaws and design errors. Thus, as already pointed out, they shall not be used to predict the chance of on-orbit failure or success but rather be applied to support the selection of different design alternatives, considering their applicability as reasonable after the spacecraft survived the infant mortality zone. The same holds true for **PRISM** and IEC 62380. PRISM allows to incorporate one's own reliability testing and part screening results, and thus obtain more realistic results for the prediction [36]. While PRISM was developed in response to obsolescence and criticisms of the MIL-HDBK-217F, IEC 62380 is a reliability prediction method based on the French Telecommunications standard RDF 2000 and is widely used in the automotive industry, but is also outdated since its underlying data were collected between 1992 and 2001.

In later design stages, when a design is known and flaws or a weak link shall be found, one can apply the second group of reliability prediction methods, targeting dependability of the system. Again, only a limited sample of methods can be presented within the scope of this work. **FTA** is a top-down approach widely used in spaceflight projects. As described by Johansson [132], FTA is an analytical technique, in which a system is analyzed to find all realistic ways in which it can fail or any other undesired event can occur. Using graphic models, a variety of faults ranging from part failures to human errors and software errors can be modeled, although the complexity quickly grows when doing so. Also, top-down, but traditionally used for more complex logical interactions, are **Petri nets** and **Markov models**. While Petri nets are general-purpose graphical and mathematical tools for describing existing relations between conditions and events [140], Markov models are probabilistic models that allow the adaption of the characteristics of individual components to the state of the system [132]. Although both methods are used regularly in spaceflight projects [141] [142], their complexity and knowledge needed exceeds the resources of most CubeSat teams. As the last top down approach described, **Reliability Block Diagrams (RBD)** can be used if the failure of either component will result in failure of the system. The block diagram does not necessarily describe the system's operational logic or functional partitioning, and the blocks can either be components or subsystems with a certain failure rate. RBDs are particularly useful when searching for dependencies among elements and/or deciding on redundancy options [132], [140].

FMEA and **Failure Modes Effects and Criticality Analysis (FMECA)** are widely used bottom-up approaches to analyze the impacts of failures of components to the operability of the system, and can be based either on a hardware or a functional approach. The former describes the consideration of actual hardware failures in the model, while the latter describes the loss of function, often in earlier design stages. The failures are ranked using a so-called Risk Priority Number (RPN), which itself is defined as the product of severity, occurrence and detection [140]. "Severity" means the severity of each effect of failure, "occurrence" the likelihood of occurrence for each cause of failure, and "detection" the likelihood of prior detection for each cause of failure, all of them typically on a scale of 1–10 [140]. The RPN can be extended

by a so-called item criticality number, describing the sum of the failure mode criticality numbers for the item. Thereby the failure mode criticality numbers themselves are calculated from [140]:

$$C_m = \beta_{cp} \cdot \alpha_f \cdot \lambda_p \cdot t \quad (15)$$

with C_m being the criticality number for one failure mode, β_{cp} the conditional probability of loss of function or mission, α_f the failure mode ratio (for a specific item $\sum \alpha_f = 1$), λ_p the item failure or hazard rate and t the operating or at-risk time of item [140]. Using FMEA and FMECA, failure modes and the causes and effects of those failures can be identified. As an input for the failure or hazard rate, field data or data from handbook-based reliability prediction methods can be used. As an output, FMEA and FMECA can be utilized as input for FTAs and RBDs. As the method is very time-consuming, and the analysis is limited to single failures [132], it is questionable if it can be used in small satellite and CubeSat projects in order to reduce infant mortality and DOA rates. To summarize, it highly depends on the system, the intended objective, and the available resources what kind of reliability prediction method fits best for a project. As a guidance, Table 6-2 provides a comparison of handbook based reliability prediction methods and Table 6-3 shows strengths and weaknesses of the most common reliability prediction approaches. Both tables are depicted in Appendix B. As further guidance, we will now have a look at a few examples for reliability prediction methods being used in space projects and their adaption for that purpose. Reliability data obtained from space missions will show us the extent of which these data can corroborate the models⁴².

In 1976, Conrad [143] proposed a reliability estimation model in which two failure rate models were combined to consider the then already known infant mortality period, in his case chosen as the first two months after launch. Binckes [144] reported in 1983 on the past and present procedures for reliability estimation of the INTELSAT series of satellites. He showed mathematical reliability models and the INTELSAT IV Reliability FORTRAN computer program being used to achieve the 7-year design goal for the satellite. Overall, a reliability of 0.69 was calculated for a seven-year mission and a probability of 0.37 of all 12 transponders operating at the end of 7 years [144]. All seven spacecraft exceeded their design lifetimes and were retired from active service in the meanwhile, the latest one, Intelsat IV F-1 in October 1987 after more than 12 years of service [145]. As a side note: one of the satellites of the series was lost in a launch failure of an Atlas rocket in 1975. Including this would lead to a total reliability of the INTELAST IV series of 0.875.

The already in Subsection 2.1.3 mentioned data obtained by Hecht & Hecht [54] showed the limits of the MIL-HDBK-217F approach as early as in 1985. They reported a factor of at least 2 between on-orbit data and data from the handbook-based prediction models, with the prediction models being too conservative. Based on their observations, they proposed a modified reliability model, in which the reliability of the mission is composed of two factors. The first factor is a constant failure rate of random hardware failures, based on an exponential model while the second factor is based on a Weibull model, taking into account design errors and environmental factors. They also presented a second model, in which they simply multiplied the results of the MIL-HDBK-217F method by a space environment factor of 0.5 [54]. In 1992, Ebeling [94] reported on a parametric estimation of reliability prediction parameters, taking into account the operating environment, reliability growth over time and technology innovations. He presented that several different types of aircraft showed an average improvement factor of 0.137, leading to a doubling in reliability of a typical F-15 aircraft within 12 years. He also reported on strong evidence of past work supporting decreasing failure rates in spacecraft subsystems (shape factor $\beta = 0.311$) [94].

In 2005 Marin & Pollard [146] first showed in-the-field experience of the FIDES reliability prediction method at Raytheon. After analyzing failure rates for circuit card assemblies, they reported on comparable results of the observed failure rate to the predicted failure rate using FIDES, while MIL-HDBK-217F predictions

⁴² Again, this is not an exhaustive list of examples. Rather, only a few examples are used to emphasize key findings.

being too conservative [146]. Using a constant failure rate through MIL-HDBK-217F, Zahran, Tawfik & Dyakov [147] presented a reliability prediction of 0.974286 for a LEO micro satellite power subsystem with a mission duration of five years. As we have seen before, it is questionable if studying only the random hardware failures is sufficient for a high-risk project such as a microsatellite, and thus if the estimation is a realistic one. In 2011, Burke and Evans [148] reported on wear-out items in electric motors in space. Out of data from past missions they found a shape parameter of $\beta = 1.2$ and a scale parameter of $\theta = 46,158$ cycles. These parameters were subsequently used in a Weibayes approach to produce an estimate of the reliability of a similar mechanism. For their specific application they noted that it is recommended to calculate future life expectancies using the Weibull distribution with $\beta = 1.2$ instead of the exponential distribution of MIL-HDBK-217F and other handbooks [148]. Wu, Yan and Xie [149] used FTA and Petri nets to analyze a solar array mechanical system in 2011 and continued their work in 2012 [150]. In 2013, Witt, Kennedy, Baetz, Mohr & Eickhoff [151] reported on a failure-aware system model of the Flying Laptop microsatellite. They developed a SysML profile to model the propagation of effects and failures and performed analyses of the imaging system and the communication system of the satellite. With the help of their tool they were able to identify potential critical failures (not receiving telemetry or receiving invalid telemetry in the system) and subsequently resolving them [151]. In the same year Kaminsky, Gallo and Evans [152] showed a different approach for reliability estimation for the deployment of the James Webb Space Telescope's sunshield. In a Bayesian prediction model, heritage data of 45 years of spacecraft launches and artificial test data were used to empirically predict the chance of for a sunshield deployment anomaly in space [152].

Independently from classical spacecraft projects, Bianchi announced a new ESA project targeting "New Reliability Prediction Methodology Aimed at Space Applications" in 2016 [70]. He reported that all current data sources for reliability prediction do not reflect the improvements in component quality. Furthermore, he stated that on-orbit performances have shown that a significant amount of on-orbit failures were due to not random failures and thus not covered by current reliability prediction methods. All prediction methods seem to be largely conservative, potentially reducing cost effectiveness and performance. In his presentation he showed that based on on-orbit data, FIDES might be a better choice for reliability prediction than MIL-HDBK-217F. While the ratio between predicted and observed reliability is less or equal two for FIDES, it is five for the MIL-HDBK-217F approach [70]. In the same year Davenel [153] reported on the motivation for the use of FIDES in space applications. He identified weaknesses in other common prediction methods, showing the aforementioned obsolescence problem of IEC 62380 and MIL-HDBK-217F as well as the limited adaptability of the PRISM/217Plus⁴³ approach. He further presented field return data, showing a factor of nine⁴⁴ between predicted and observed reliability for military equipment using MIL-HDBK-217F and 2.8 using PRISM/217Plus [153]. Also in 2016, Pearson, Callen, Blanquart, Bourbouse & Gajewski [154] presented their results on an ESA funded study evaluating reliability prediction data sources. They showed that on-orbit feedback from the Spacecraft Computer Unit (SCU), the central computer of the Eurostar 3000 satellite bus, experienced a failure rate 0.9 times the failure rate predicted by FIDES, and 0.25 predicted by the MIL-HDBK-217F approach. They concluded in their report that for Electrical, Electronic and Electromechanical (EEE) parts FIDES appears to be the best approach [154]. The same conclusion was reached by Bourbouse et al. [155], showing additional data from Travelling Wave Tube Amplifiers (TWTAs) of the Eurostar 3000 satellite bus. The MIL-HDBK-217F estimated failure rate of 900 FITs (FIT = Failure per billion hours) differed widely from the on-orbit experience of 200 FITs. They recommended FIDES for the majority of electronic parts over MIL-HDBK 217 and 217Plus [155]. Table 6-4 in Appendix B depicts the full list of recommendations. Huang, Loman, Andrada & Ortlund [156], while presenting their Bayesian reliability

⁴³ 217Plus is a spin-off of PRISM. It uses the same modelling methodology, but has increased the number of part type failure rate models [140].

⁴⁴ This means that the reliability was nine times better than predicted, or in other words, the observed failure rate was a ninth from the predicted value.

approach, also reported a ratio of four between on-orbit experienced and predicted reliability of Travelling Wave Tubes (TWTs) in 2016.

In 2017, Carton, Giraudenau & Davenel [139] reported on the results of the REX study by the French military, in which feedback from land, air and sea applications was collected for 24 months. They showed large dispersions between the predicted values by MIL-HDBK-217F and the observed values, and better predictions when using the FIDES approach. Depending on the component, FIDES was off by a ratio within one and three, which was concluded as being satisfactory by the authors of the study [139]. Finally, also in 2017, Höfner, Vahl & Stoll [157] presented a top-down approach for reliability analyses adapted from commercial aviation. They used component importance analyses and identified AOCs and the Electrical Power System (EPS) as their most vulnerable subsystems, and presented reliability estimations of 0.9999, 0.9945 and 0.9997 for three different satellite design options [157].

To conclude, different from the expectation of some program managers, reliability prediction is a process with uncertainties, but is often misused as being as accurate as the prediction of physical parameter such as mass or power. All handbook based methods assume the system reliability as the product of all component reliabilities, and that part failure data from the past can be used for future, in reality often different designs [134]. As we have learned, historic data [54] oppose the view that system-level reliability is dominated by random part failures. Furthermore, improved component quality of COTS electronics, an increasing complexity as well as increasing amounts of software have further eroded this point of view. System-level factors, such as design and engineering errors as well as software flaws dominate today's breakdown of on-orbit failure. As O'Connor noted, failures are ultimately caused by people, and the ways in which they are managed, and that is the major determinant of reliability and safety [134]. Reliability prediction can serve as tool to trade off different design options, assuming only constant failure rates and prevention of the above mentioned major causes of failures. Current models ultimately cannot predict the overall on-orbit reliability with a sufficient accuracy as they currently only allow to cover the random error and partly the wear-out phase of the bathtub curve. A study by the Reliability Analysis Center showed that 78% of failures of electronic systems arise from non-component causes [158], and many of them, as we have seen, emerge early in the mission. The infant mortality portion of the overall reliability is hard to predict since it depends on a multitude of different, mission-specific aspects. Nevertheless, to achieve reliable systems, other methods to prevent early failures can be used for spaceflight projects, and this leads us to our next subsection, focusing on reliability assessment.

2.2.2 Reliability Assessment in Space Missions

We defined the assessment of reliability as conducting tests on part-level, subsystem-level or system-level in a relevant environment or getting feedback from on-orbit⁴⁵. Relevant environment means that the system must be tested at temperature and pressure levels similar to the space environment, and that it also must withstand launch loads. In traditional systems, standards are being used to assure that a space system withstands these loads, but rather as a go/no-go prove than a statistical assessment. We will summarize selected standards in the beginning of this subsection. Contrary to terrestrial systems, as we have already seen, space systems are in most of the cases one-of-a kind, one-shot systems. Thus, lacking statistical relevant numbers of items, many traditional assessment methods cannot be used for space applications. Accelerated life testing is an approach that could be utilized to assess the reliability of parts or assemblies before launch. Of course, this method can be rarely used on system-level, as most of the space projects cannot afford a statistically relevant number of spares. Therefore, the second part of this subsection will

⁴⁵ Another definition sees reliability assessment as the process needed to evaluate the value of reliability to the system's stakeholders, and the associated costs to that [130].

deal with the applicability of accelerated life testing for spaceflight. In the third part, we will summarize key points of reliability growth, which can be used to evaluate results of reliability assessments.

The European Standard for Testing in Space Engineering, EN 16603-10-03:2014 [159] defines the system-level tests needed to prove that a space system is designed and built suitable for space (and launch) environment. The standard originates from the European Cooperation for Space Standardization (ECSS) Standard ECSS-E-ST-10-03C [160]. The detailed steps needed are depicted in Figure 2-26 and will not be further described in this work. As already pointed out, each test is more or less a pass or fail test, in which the reliability of the system is not assessed. A functional and performance test is included as one of the last steps and has the goal of verifying the complete function of the spacecraft and the compliance of the spacecraft's performance to its specifications, both in relevant environment. Further test specifications, not mentioned in the overview, describe a life test, a burn-in test and a mission test for space systems. The life test is described as demonstration of the ability of the spacecraft to withstand the maximum number of operational cycles during its planned lifetime in the specified environment. For the burn-in test, temperature and operating time needed to eliminate infant mortality shall be agreed on with the customer. And finally, the mission test shall demonstrate the ability of the spacecraft to carry out critical and main operations covering the events of the actual flight sequence [159]. Although all of these tests seem reasonable and a qualification of the space hardware to the space and launch environment is inevitable, the feedback from reliability data of past missions and the high rate of infant mortality cases for CubeSats raise the question, if additional tests and methods to uncover engineering mistakes are needed. Furthermore, no description of the duration or the statistical evaluation are given for the full functional & performance, life, burn-in and mission tests. As mentioned before, these tests can thus rather be considered as go/no-go evaluation than a statistical reliability assessment.

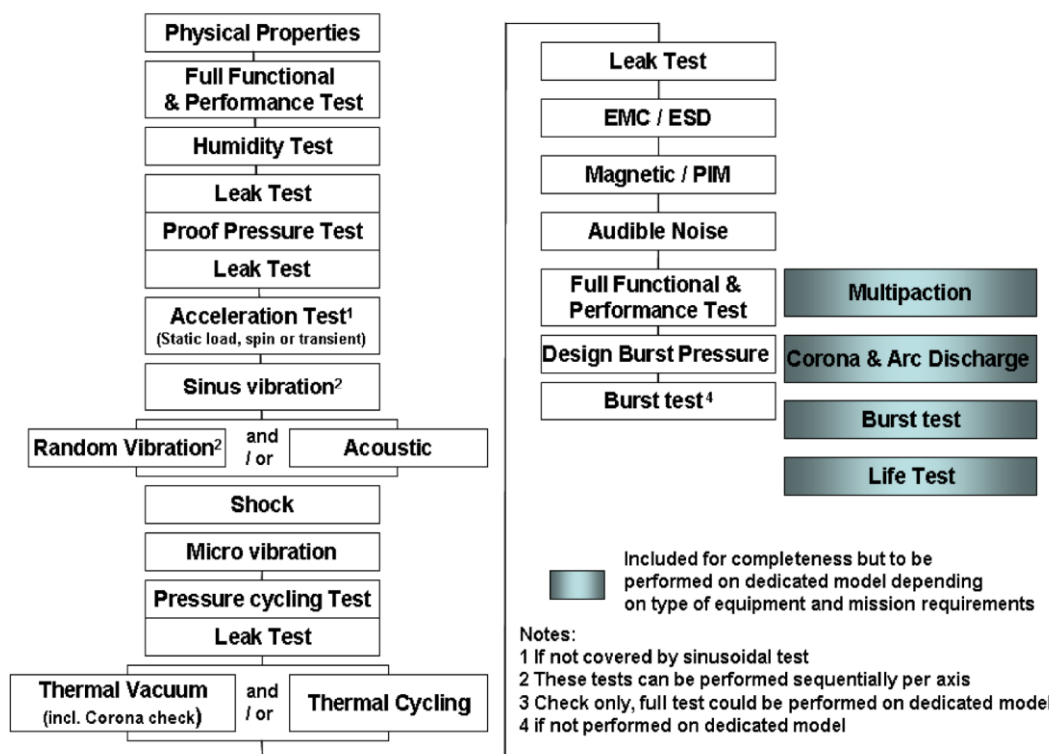


Figure 2-26: System-level test sequence for spacecraft. Source: [159]

Apart from the European Standard, there is also an international standard in place: ISO 15864:2004 deals with space systems and general test methods for spacecraft, subsystems and units [161]. The ECSS provides several standards and handbooks on this topic: the standard ECSS-E-ST-10 deals with consistent

application of ground testing requirements to allow proper qualification and acceptance of space products [160], while the standard ECSS-E-ST-10-02C defines fundamental concepts of the verification process [162]. The handbook ECSS-Q-HB-30-08A describes data sources and respective methods for reliability prediction of components [163] (more suitable for the prior subsection) and the handbook ECSS-E-HB-10-02A provides additional information for the application of the standard ECSS-E-ST-10-02C [164].

In the US, similar standards and handbooks are available for space applications: the DoD Standard Practice for Product Verification, Requirements for Launch, Upper Stage, and Space Vehicles [165], the DoD Handbook for Design, Construction, and Testing Requirements for One of a Kind Space Equipment [166], the NASA Technical Standard for Payload Test Requirements [167] and the NASA Goddard General Environmental Verification Standard (GEVS) for GSFC Flight Programs and Projects [168]. Additionally, there are numerous guidelines and standards of larger private companies and entities on how to assess and verify their space hardware before launch. The Aerospace Company is one example for that, having published their own Flight Unit Qualification Guideline [169]. Overall, most of those guidelines, handbooks and standards focuses on the verification of functions or parameters against known values. As aforementioned, this is inevitable for the qualification of hardware for space and launch environment, but it might not be sufficient to uncover engineering mistakes or failures that need tight coupling and complex interactions (mostly on system level) to occur.

Usually, the V-Model is a widely used model for system development in spaceflight (see Figure 2-27). Going down to part level (not shown in the figure), each step of hardware development is preceded by specific requirements, and followed by verification of these requirements after design. Thus, in an ideal case, little to no inconsistencies between the subsystems should exist when verifying the system. As we have seen, modern complex systems, with heavy reliance on software⁴⁶ make it almost impossible to know all interactions and related failures beforehand. Especially when resources, time and experience are limited, the verification process cannot fully cope with all possibly occurring failures on system level beforehand. Thus, as already pointed out, small satellites and CubeSats should focus in particular on system level functional tests, as those tests will be the only chance to uncover engineering flaws and other not-part related early failures that would otherwise lead to infant mortality.

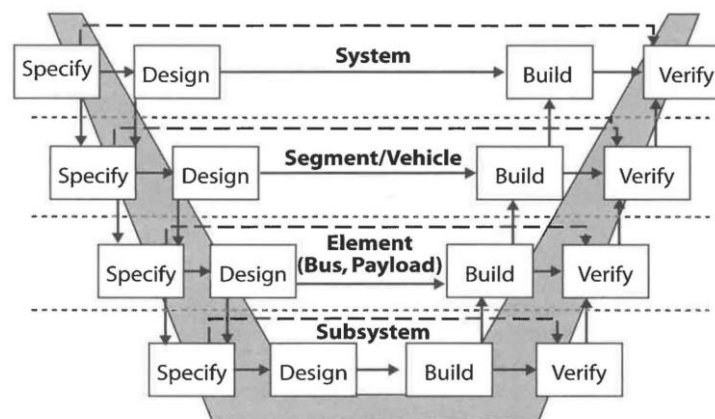


Figure 2-27: The V-Modell of development and verification of space hardware. Image source: [170].

For many modern products, lifetime expectations are in the order of years or even a decade. As Escobar & Meeker [171] noted, few units will fail or degrade in a test of practical length at normal use conditions, as pointed out by the example of a communication satellite by them⁴⁷. Thus, to faster acquire information on reliability, this led to the use of accelerated tests (AT), in which components, subsystems or entire systems

⁴⁶ And in many cases not-well defined requirements of this software.

⁴⁷ Service life of 10 to 15 years vs. allowed testing time of eight months [171].

are exposed to higher levels of stress. These results are then used to predict the life expectation of similar items under not accelerated (i.e., real) conditions [171]. Usually, these predictions take the number of units and the operating hours before failure into account [44], but the assumption that the acceleration factor only modifies the parameter of the failure-free time distribution, not its type, has to be carefully evaluated for every accelerated test [39]. As pointed out by Escobar & Meeker [171], two different methods of accelerated tests can be distinguished. In quantitative accelerated tests, already known failure modes shall be investigated to find the failure-time distribution or degradation distribution of items. Qualitative accelerated tests have the goal of identifying product weaknesses caused by flaws in the product's design or manufacturing process [171]. We will focus on qualitative accelerated tests in the following, which can be further categorized depending on the group of failures to be found and the magnitude of acceleration. As Collins et al. noted [172], many terms used in accelerated testing are highly interchangeable, so many of the following terms will be used in different ways by practitioners. The first distinction can be made between accelerated life testing (ALT) and highly accelerated life testing (HALT), although, according to Collins et al. [172], no precise definition exists of the difference between "accelerated" and "highly accelerated" in terms of time or variations of variables. The differences are in the data collected and what is done with it. While ALTs have the goal to collect data that enable predictions about the service life of a product, HALTs are used to identify design weaknesses or engineering flaws within every phase of the bathtub curve [172]. Thus, ALTs could also be grouped into the methods of quantitative accelerated tests, while HALTs would be a qualitative way of accelerated testing.

Within the HALT method, one can further distinguish between tests to find and fix production flaws (i.e., infant mortality), called highly accelerated stress screening (HASS) and tests to identify weak links of the design, which are summarized as HALTs. Figure 2-28 shows the phases of the bathtub curve and the appropriate accelerated tests to identify weaknesses associated to the phases. The majority of consumer products such as cellular phones or laptop computers nowadays undergo some kind of HALT/HASS before market exposure or delivery [172]. Thus, the environmental tests required by the presented testing standards and handbooks could be seen as some kind of HASS, although the acceleration factors or testing time might not be sufficient for that purpose. As Maurer [36] noted, power burn-in tests are used in commercial production lines of electronics to sample out items of the production lot so weak that they would fail after a short service life. These tests, usually last for a week or ten days (168–240 h) and will cause only little fallout if the product is matured and the manufacturing technique is established [36]. On the contrary, for many spacecraft, the design is the first of its kind, and sometimes even the production process of hardware itself is novel. Thus, as presented before, many failures will emerge early in life as supported by on-orbit reliability data.

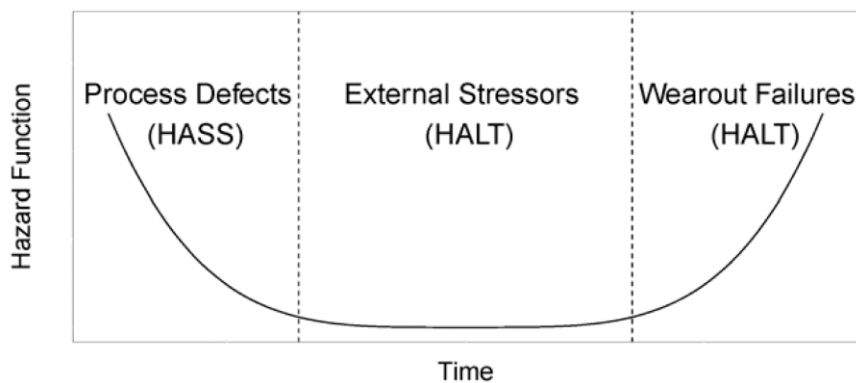


Figure 2-28: Highly accelerated tests used to identify weaknesses of different product phases. Source: [172]

In all accelerated test methods, different acceleration models are possible, depending on the item under test, the physical relationship of its properties to reliability, and the end use environment. Escobar & Meeker [171] describe an increase of the use rate of the product, an increase of the intensity of the exposure to

radiation, an increase of the aging rate of the product, and an increase of the level of stress (e.g., amplitude in temperature cycling, voltage, or pressure) under which test units operate as possible options. The simplest and most important models are the Arrhenius model and the Eyring model, both of them using exponentially distributed failure times [172]. For the widely used Arrhenius model, the temperature is the accelerating factor, since the reaction rate of a chemical reaction can be written as [171]:

$$R_r(T) = \gamma_0 \cdot \exp\left(\frac{E_a}{k \cdot T}\right) \quad (16)$$

where $R_r(T)$ is the temperature dependent reaction rate, γ_0 the pre-exponential factor⁴⁸, E_a the activation energy, k the Boltzmann constant and T the temperature. Since the inverse of the Boltzmann constant is approximately 11,605 the acceleration factor for the Arrhenius model a_{afa} can be determined by [171]:

$$a_{afa} = \exp\left[E_a \cdot \left(\frac{11605}{T_U} - \frac{11605}{T}\right)\right] \quad (17)$$

with E_a being the activation energy, T_U the product use temperature and T the elevated temperature for the acceleration test [171]. The second model, the Eyring model, is again based on chemical reaction rate theory, but applicable to phenomena such as diffusion, corrosion or migration [172]. The acceleration of the reaction rate is an extension to the Arrhenius model and can be written as:

$$R_r(T) = \gamma_0 \cdot A \cdot \exp\left(\frac{E_a}{k \cdot T}\right) \quad (18)$$

with A being a function of temperature, depending on the specifics of the reaction dynamics, also being defined as [171]:

$$A = \left(\frac{T}{T_U}\right)^m \quad (19)$$

and with m being the Eyring acceleration factor. The acceleration factor a_{afe} of the Eyring model is [171]:

$$a_{afe} = \left(\frac{T}{T_U}\right)^m \cdot a_{afa} \quad (20)$$

Maurer [36] gives an example for an acceleration model for testing bond-wire fatigue failures in LEO space equipment. For his researched space application, the temperature in the field ranges between -30°C and 55°C , and thus the acceleration factor for a 1,000-cycle temperature test between -55°C and 125°C is 20, meaning a 1,000-cycle test represents 20,000 temperature cycles in space. Depending on the specific orbit height in LEO, this means between 3.4 and 4.2 years [36]. Kosinski and Cronin [173] reported on the HALT program at Space Systems Loral. Starting in 1996 with a part-level problem of a power control unit, they described the different use cases of HALT in commercial satellite production. Although they restricted HALT to “appropriate cases”, they emphasized the benefits of HALT over traditional MIL-spec qualification tests: while in traditional tests the goal is to pass the test or in some cases explain away a failure, if one occurs, the goal in HALT is to try to force failure to identify weak links and design flaws, and ultimately to make the product more robust prior to moving to production. This is, according to Kosinski and Cronin, especially important when introducing new technologies and complex designs for use on commercial satellites in order to improve reliability over time by stimulating, and afterward correcting failure modes [173]. The limits

⁴⁸ An empirical relationship between temperature and rate coefficient.

and pitfalls of accelerated testing lie in the involved extrapolation and the deep knowledge needed on the underlying physics or chemistry of the failure mechanism. As Escobar & Meeker [171] pointed out, any extrapolation requires a physical or chemical explanation of the accelerating variable on the failure mechanism. More often though, empirical relationships are used as justification. Also, accelerated testing is usually carried out for one particular, not multiple failure mechanism. It is then possible and realistic that the different failure mechanisms will be accelerated at different rates leading to incorrect estimations [171]. Meeker, Sarakakis and Gerokostopoulos [174] provided an extensive overview on many pitfalls of accelerated testing, which can be broadly grouped into pitfalls caused by statistical misconceptions, and pitfalls caused by the naive application of accelerated test methods without good knowledge of the underlying physics or chemistry of the failure mechanism [174]. Furthermore, there is a misconception of accelerated testing as a method to predict product reliability, rather than an iterative process of finding and removing defects [172]. On the next pages, we will discuss reliability growth modelling as a method to predict product improvement and remaining failures out of assessed reliability data.

Reliability growth is the process of removing initial design and manufacturing flaws from a system and tracking that process via mathematical growth models. As defined by Hall [175], the prototypes of systems are subjected to environmental, mechanical, thermal or electrical stresses similar to those encountering in the operational environment. Growth means that there is an increase in the true but unknown reliability over time, since discovered failures are resolved with corrective actions, either in hardware, software or human factors [175]. Hall distinguishes between reliability growth projection, used to quantify the hypothetical reliability of the next configuration of the system, reliability growth planning for managing resources and timelines associated to the test program, and reliability growth tracking, used to measure the reliability improvement effort through the development of a system [175]. Reliability growth always works with the assessed reliability of the system through testing, and in most models it is assumed that only a certain rate of failures can be fixed, quantified by a Fix Effectiveness Factor (FEF) [176], which is usually assumed to be around 0.8 [175]. Historically, although some work started already in the 1950's, the 1962 paper of Duane [177] marked an important first step for reliability growth modelling. He found that different systems undergoing reliability testing and improvement at General Electric experienced a linear relationship of the failure rate vs. cumulative operating hours, when plotted in log-log style [175], [176]. The relationship can be seen in Figure 2-29, in which the reliability growth of several different systems found by Duane are plotted. The value α is called the improvement factor, and it describes the overall rate of reliability improvement of the system under test [175].

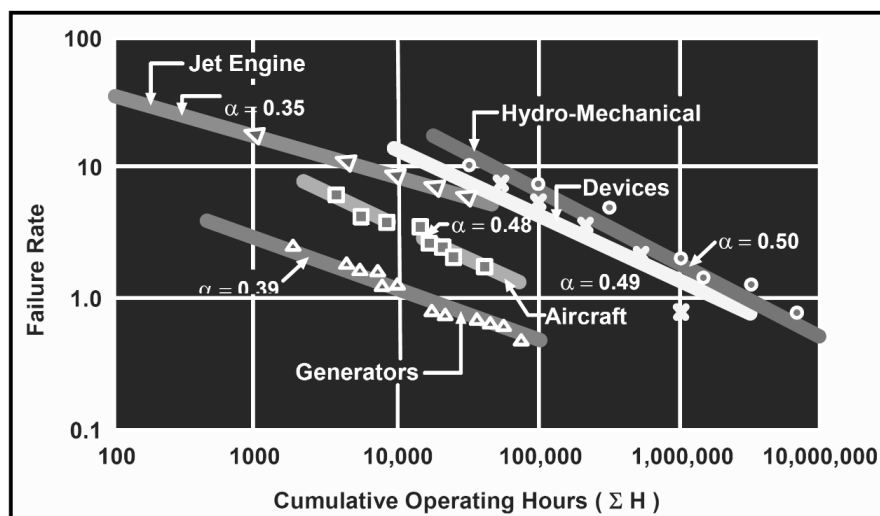


Figure 2-29: Reliability growth experienced in different systems under test. Original data of Duane's publication. Adapted from: [176]

As Birolini [39] noted, reliability growth itself cannot distinguish between failures due to engineering errors and manufacturing flaws, and random failures, although engineering errors and manufacturing flaws should emerge early in life and random failures should occur at a constant failure rate over the whole lifetime. It is thus necessary to conduct a thorough root cause analysis of every failure, and subsequently correct it. In commercial products, a reliability growth program is a cost-efficient way of reliability improvement by eliminating the cause of design and production weaknesses. It is mostly performed during prototype or pilot production and not in series production. Figure 2-30 depicts typical reliability prediction and reliability growth curves of a commercial product [39]. This could also be applied to spaceflight applications, though in most cases, and especially for current small satellites⁴⁹, the single-item characteristic would result in just one growth curve on the FM/EM.

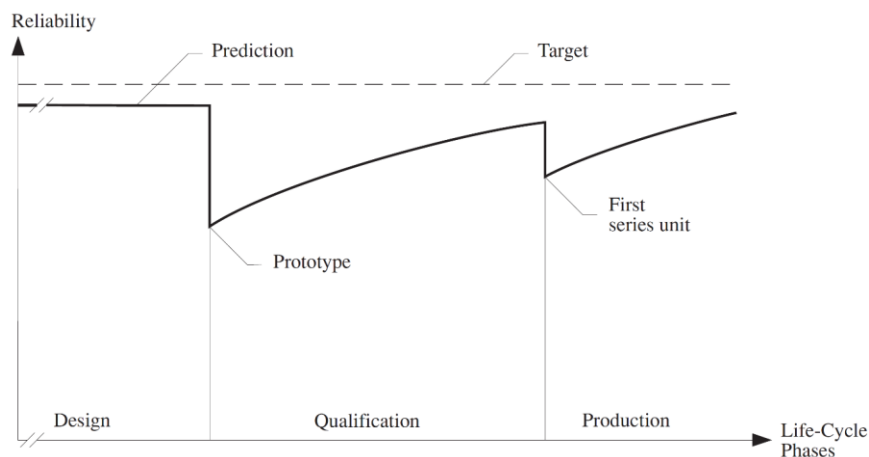


Figure 2-30: Reliability prediction and reliability growth for a serial-produced, commercial item. Source: [39]

In reliability growth models, different from other reliability models, the statistical evaluation is not restarted after the system was changed or modified (due to failure correction) [39]. Also, it is important to understand that different environments at different developmental stages can affect the outcome of reliability tests and thus also the reliability growth curves. An example would be a first reliability test at low temperatures, followed by corrective actions, and a second test at high temperatures. Although the failure rate is expected to decline because of the corrective actions, it might be higher than in the first test due to the higher temperatures. In some cases, separate reliability growth curves for different environments or use cases might even be needed [138]. In the following, we will summarize important reliability growth projection models. Later we will also look at software reliability growth models, and then conclude this subsection with examples of reliability growth modelling in spaceflight.

Based on the findings of Duane, Crow [176] showed that the reliability growth failure times followed a non-homogeneous Poisson Process (NHPP), equivalent to a Weibull process, in his 1972 paper [172]. The failure rate for the Crow-model is [138]:

$$\lambda(t) = \mu \cdot \beta_g \cdot t^{(\beta_g - 1)} \quad (21)$$

with μ being a scale parameter, and β_g the reliability growth parameter. $\beta_g < 1$ means reliability growth, $\beta_g > 1$ reliability decay and with $\beta_g = 1$ the model can be reduced to the homogeneous Poisson process model [138]. Crow observed β -values of around 0.5. The model was subsequently referred to as the AMSAA model, the Crow model, or the AMSAA-Crow model, in which AMSAA stands for the Army Materiel System

⁴⁹ Of course, with the important exception of series-production of small satellites and CubeSats, such as the constellations of OneWeb, Planet and Skybox, where an approach similar to the commercial terrestrial industry could be applied.

Analysis Activity. It is the most frequently used reliability growth model, and in an update of the model in 2011, AMSAA suggested values of β_g in the range of 0.3–0.75, depending on the level of commitment to reliability improvement and the type of system under development [138]. Crow extended his model in 2005 with the option of delayed or non-delayed fixes [175]. In 2011, he presented [176] an average β_g of around 0.7, and suggested that for complex, heavy mechanical/electronic systems β_g should be 0.73 or larger, based on his data. For the FEF he found an average of 0.69, with an FEF of 0.8 being most suitable for complex commercial electronic systems. Also, in his paper, further parameters for reliability growth projection models such as the Discovery Function, Discovery Parameter and Management Strategy Parameter are discussed [176].

Common reliability models are mostly still hardware-centric, but, as noted in a report by the National Research Council (NRC) [138], in practice many of them are used to trace system failures, which could be both of hard- and software origin. Although software failures, as we have seen, have a different statistical way of occurrence than hardware failures, the resulting software reliability growth might fit to the general reliability growth models. It has to be noted though that software systems are more susceptible to additional correction inserted bugs than hardware errors. Thus, some of the most common NHPP models used for hard- and software growth modelling will give poor inferences for software systems in development [138]. Since we have learned that current and future satellites are more and more software-centric systems, it is worth to also summarize some of the most common models for software reliability growth modelling.

In the 1979 model of Goel & Okumoto [178] a NHPP model is used to exponentially fit reliability growth to software systems. They studied their approach with data from the errors in the development of software for the real-time, multicomputer complex of the Naval Tactical Data System (NTDS). In their model, the number of estimated errors up to a time t is [178]:

$$H(t) = a \cdot [1 - \exp(-b \cdot t)] \quad (22)$$

with $H(t)$ being the number of estimated errors up to time t , a being the expected initial error content at $t = \infty$ and b being the constant error detection rate per undetected error at time t [178]. In 1984 Ohba [179] presented improved models for software reliability growth, the so-called delayed S-shaped model, the inflection S-shaped model and the hyperexponential model. While studying RADC test data and data of other origin, he found an S-shaped model most suitable for real projects, due to the increasing testing effort over time (see Figure 2-31). He also reported of “rare occasions”, in which instead of the S-shaped model a hyperexponential model was the best choice, especially when dealing with large systems consisting of a variety of different modules [179].

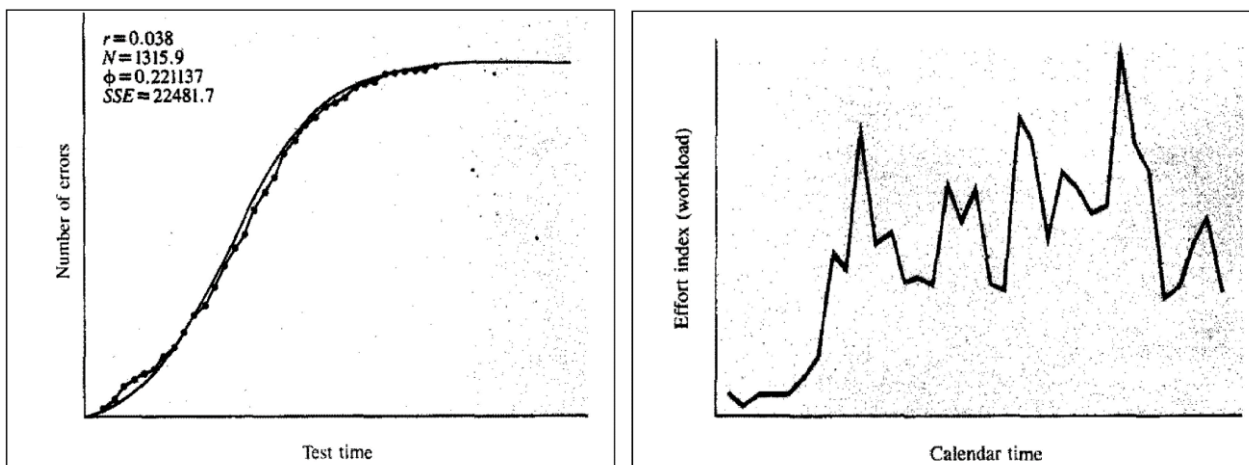


Figure 2-31: Delayed S-shaped growth model for software reliability data of a RADC project (left) and testing effort of the same project over time (right). Adapted from: [179]

In 1985 Yamada & Osaki [180] presented a modified exponential growth model, in which a nonhomogeneous error detection rate is applied and two different types of errors: easy- and difficult-to-detect errors. As the last example of software reliability growth modelling presented in this thesis, Gaver & Jacobs [181] showed an inverse Weibull function to be best fitting to their data in 2014. We will revisit some of the models in Section 4.3. The last part of this subsection will deal with examples of reliability growth applied to space projects, except for small satellites and CubeSats, which we will cover in the next subsection.

As we have seen in Subsection 2.1.3, reliability growth and saturation curves have been used for spacecraft since the dawn of spaceflight. Norris showed in the aforementioned 1976 paper [83] that both Duane and Weibull models fit the reliability growth experienced in test as well as in flight. In 2012, Strunz & Herrmann [182] showed how to use a Bayesian approach to model reliability growth data of liquid rocket engines. Besides modelling test data, reliability growth can also be used on a series of satellites, as presented by Evans, Kaminsky & Gallo [183] in 2012. They used data on the Tracking and Data Relay Satellite (TDRS) and on the Geostationary Operational Environmental Satellite (GOES) series. Overall, 327 on-orbit spacecraft anomaly reports on two generations of nine TRDS and two generations of 12 GOES satellites were analyzed for the study. Evans et al. used the before presented Crow-AMSAA model and showed a clear reliability growth over the number of satellites launched per series [183]. As we have learned before, this behavior is expected and can be explained by aforementioned extinction of systematic errors over increasing numbers of satellites of the same type. As depicted in Figure 2-32, the failure rate decreases over the number of satellite launched per series. The data were also used to predict the reliability of TRDS 13 (TRDS M)⁵⁰ before launch. The decrease in reliability between GOES 6 and 7 can be explained by a new feature implemented in GOES 7. Also GOES 8, the first satellite of the second GOES generation, experienced a higher failure rate than the satellites before, again explainable by systematic errors of a new design flying for the first time on that mission. Interestingly this deviation cannot be seen in the TDRS data. Lastly, the jump between GOES 11 and GOES 12 was attributed to a new type of instrument carried for the first time by GOES 12 [183].

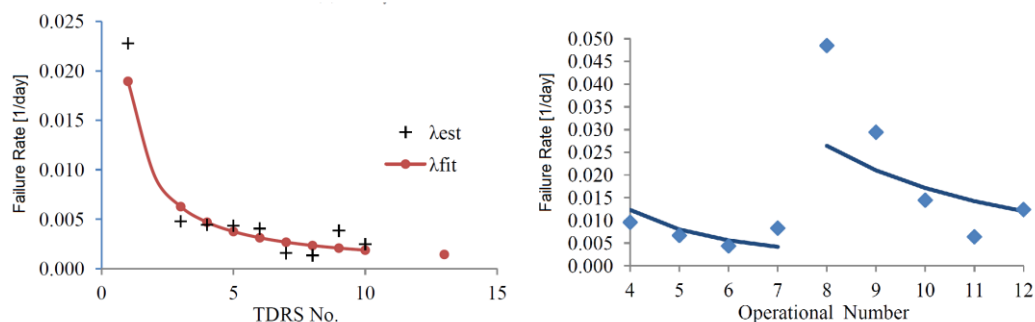


Figure 2-32: Reliability growth of the series of TRDS (left) and GOES satellites (right). Adapted from: [183]

Evans et al. furthermore presented the cumulative number of failures on the GOES satellites. As can be seen in Figure 2-33 and Figure 2-34 the decreasing failure rate also holds true for this type of satellite. For later satellites of each generation, a decrease in early failures (extinction of systematic errors) can be noted. As already pointed out, new technology was flown for the first time on GOES 7. Lacking the original failure data, we can speculate that the increase in early failures of GOES 7 was also caused by systematic errors when flying a system, a subsystem, or a part for the first time in space. The same decrease in total failures over time can be noted for the second generation of GOES satellites, as depicted in Figure 2-34. Again, a small increase can be observed for GOES 12. As with GOES 7, this could also be caused by a new instrument flown for the first time on GOES 12, but the sharp increase in failures is not fully explainable without the underlying data. The overall increase in failures of the second generation of

⁵⁰ TDRS M was launched in August 2017.

GOES satellites is explained by Evans et al. by more complex satellite design and functions implemented [183]. This also agrees with the complexity vs. failure relationship mentioned earlier.

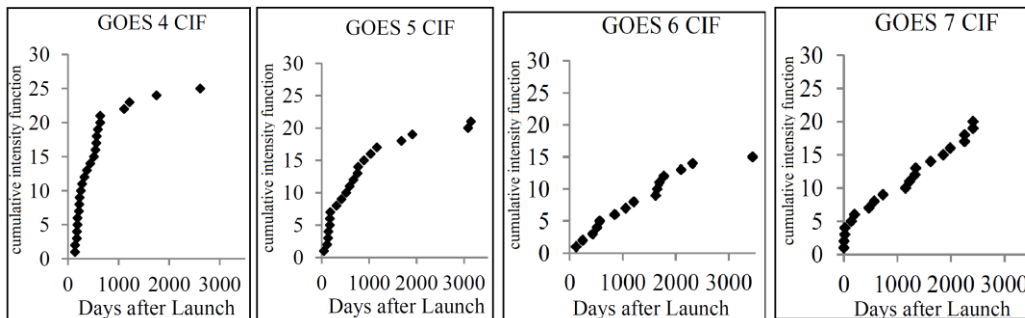


Figure 2-33: Cumulative number of failures of the first generation GOES satellites. The increase in early failures of GOES 7 could be explained by new features implemented on that satellite. Adapted from: [183]

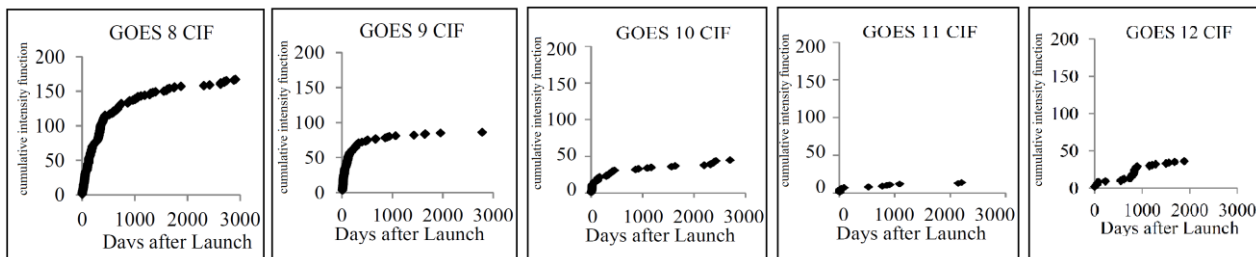


Figure 2-34: Cumulative number of failures of the second generation of satellites of GOES satellites. The overall increase in failures with respect to the first generation can be explained by a more complex satellite design and functions implemented. Adapted from: [183]

For software reliability growth models, Sukhwani, Alonso, Trivedi & McGinnis [184] presented results of an analysis of NASA space flight software. For their study, they studied anomaly reports for the development and testing phase of on-board software of launched space missions over several releases, using data of the tracking tool of GSFC. For most releases that experienced growth they found NHPP and S-shaped models to be best fitting. As can be seen in Figure 2-35, the anomaly count decreases over several releases of the software. For releases 2.0.0b and 4.0x-4.2x growth was found, while 2.0.0a, 3.x and 4.3-4.7x experienced no trend. Only for 1.0.0 and 2.3.x a decay in reliability over time was found. Sukhwani et al. explain the decay in 1.0.0 with the test team rushing towards the deadline for handing over Build 2.0.0a to the test team. Similar to release 1.0.0, they also explained the decay of releases 2.3.x with a lot of Build Integration activity for the upcoming FlatSat testing [184].

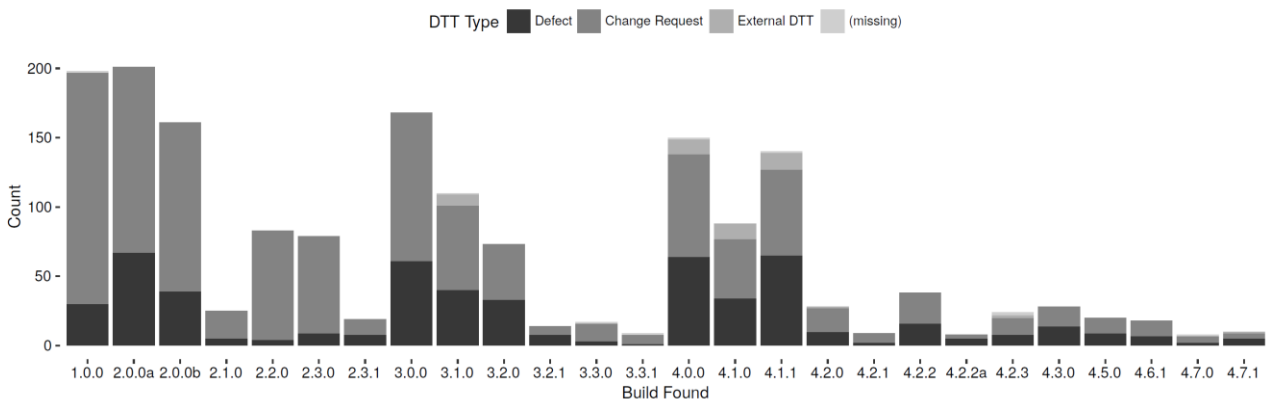


Figure 2-35: Anomalies found in releases of on-board software for a NASA mission. Releases up to 2.00a were operated on COTS hardware, up to 2.3.0 on the Engineering Model (EM), up to 4.3.0 on the FlatSat and after that on the Flight Model (FM). Adapted from: [184]

Finally, Cho, Jang and Park [185] used the Crow-AMSAA model for reliability growth planning and tracking of 27 Hybrid DC-DC Converter of GSFC. They pointed out the necessity to use reliability growth to accomplish reliability for complex systems, and showed that their estimated failure rate effectively identifies the current reliability of a target system [185].

To conclude this subsection, assessing the reliability of a system has fundamental differences to predicting the reliability of the same system. Although the data of the former can be used as input for the latter, reliability assessments generate a more realistic reliability estimation of the system, if applied correctly. On the other hand, a study by the NRC in 2015 [138] pointed out that the majority of system acquired by the US Army showed clear gaps between the reliability estimated by development testing and the reliability in operational testing, with many systems failing in operational testing. The Chief Scientific Advisor to the Director of Operational Test and Evaluation attributed this to false predictions stemming from the wealth of reliability growth models available, and he recommended reliability growth modelling to be only used for prescribing test duration required to reach a level of acceptable reliability, instead of estimating in-the-field subsystem or system reliability [175]. Thus, for small satellite and CubeSat development, reliability growth modelling could help provide reasonable estimates of testing time needed to mature new designs sufficiently before launch. While doing so, varying loads due to environmental testing as well as incrementally increased functionality can inflate or alter reliability growth results, thus this has to be carefully dealt with [138]. The TLYF approach, described in more detail in the next subsection, is one way how these alterations can be limited in spacecraft development. Being mostly one-of-a-kind and one-shot items, reliability growth can be utilized on several stages of the spacecraft under development (EM, FlatSat, FM), but rather seldom on multiple items of a series of satellites. Lastly, as Meeker and Hamada pointed out, reliability can only be assessed directly after a product has been in the hands of customers for a significant amount of time [172]. Yet waiting for that field experience to prove or disprove reliability, in our case the spacecraft being on-orbit, would be a costly option if not reduced to a certain level of risk by a reliability assessment program on ground to eliminate systematic errors and engineering flaws. In traditional space applications, multiple methods are applied to assure reliable space missions, some of them applicable for small satellite and CubeSat missions, some not. The following subsection summarizes the most important methods and evaluates their suitability for low-cost and fast-delivery small satellite and CubeSat programs.

2.2.3 Reliability Assurance in Space Missions

Reliability assurance is a broad term that loosely groups practices used in space programs to guarantee a certain reliability of the spacecraft. Hurley Jr. & Purdy [107] presented nine different methods how reliability is assured in modern space projects: Good Design, Thorough Testing, Flexibility and Margins, Redundancy, Use of Mass-Production Components, Reliability Analysis, Rigorous Manufacturing combined with Quality Assurance and Processes, Mission Simulation and Training, Constellation Design and Launch-On-Demand. These methods range over the whole product lifecycle and we will address the most important ones and comment on their applicability in the small satellite/CubeSat domain in the following. Historically, one of the strategies to achieve reliable spacecraft was developed by the manufacturers of the early Intelsat spacecraft [19]. JPL later reported on its secrets to long-life spacecraft [186], as many of the JPL-produced spacecraft vastly exceeded their lifetime. They presented wide performance margins, a strong environmental test program, more than 1,500 hours of system operating time prior to launch, block redundancy, software design flexibility and minimum Class B screened⁵¹ parts as their way to achieve reliable and persistent spacecraft. As a side note, JPL reported that all of their missions had some workmanship failures detected during testing that would have been mission limiting [186]. However, in today's commercial space industry, cost-efficiency prevents some of the strategies used by JPL for their interplanetary missions. Satellite manufacturer nowadays achieve reliability targets by using standardized platforms for different

⁵¹ As we will see in Section 2.3, Class B denotes parts screened for high reliability military applications [187].

applications, verify those satellites in a cost-effective manner with only a reasonable number of tests [19], and then can eliminate remaining systematic errors in forthcoming items of the series. For small satellites and especially CubeSats that approach also works in the case a constellation or series is planned. For university built CubeSats this approach works rarely. If little time is between succeeding programs and knowledge drain (or different mission goals) are not preventing the use of heritage, it could work somehow, but in all other cases, different strategies have to be found to cope with the risk of early failures and systematic errors.

Good design and redundancy have always been a cornerstone of spaceflight. As we have seen, instead of wide assumption that parts are the source of failure in spacecraft systems, design errors cause more harm than part level issues. Of course, this could be attributed to the high level of part quality common in today's spaceflight projects, but past missions also showed this pattern. Design errors often involve a multitude of parts or subsystems, linked to the tight coupling and complexity of spacecraft. While the Keep it simple, stupid (KISS) principle is also true for spaceflight, mission goals sometimes demand complex hard- or software. Heritage, another reliability assurance strategy of spaceflight, can then be used to circumvent the risk of design errors due to novel applications, if possible. Although heritage can decrease the risk of the mission, heritage designs must be requalified in any case for new applications, as seen before. The loss of the Contour mission⁵² or Ariane V Flight 501 are examples of heritage gone wrong [69]. Thus, in general, spacecraft design always involves a number of judgement calls that have to be made by engineers and managers involved [19]. Should the mission rely on heritage design and use an old, well-qualified electronic component or a heritage design and thus sacrifice power savings, processing speed, or flexibility that could come with novel electronics? Or the other way around, should a mission manager take the risk of implementing a new design if only moderate power savings and a slightly better performance can be achieved? What if a part becomes obsolete? All these questions can only be partly solved from a mission point of view. If power savings are mandatory or a mandatory function can only be provided by a new design, then the judgement call is sometimes easier, but more often it is a multi-dimensional problem involving many trade-offs. Small satellites and CubeSats could be one way to circumvent this, testing new designs fast and on a footprint and providing the necessary assurance for bigger missions.

Redundancy in spaceflight can either be applied on system, subsystem or part level. In general, many applications achieve their reliability and availability goals often with the help of redundancy. For redundancy, common cause failures in production and design have to be prevented, thus parts, subsystems or systems should be designed and manufactured independently from each other [39]. This could also be applied to software, in which redundancy normally is not a measure to prevent failures, since, as we have seen, software only fails due to systematic errors that would also emerge in the redundant copy if not developed separately⁵³. However, as we have learned from Leveson, humans tend to make similar mistakes, thus there are also limits on what to achieve with software redundancy⁵⁴. As we have seen, the failure rate of a series system consisting of independent elements is theoretically just the sum of the failure rates of these elements. For redundant systems, we can distinguish between hot redundancy (full load), warm redundancy (reduced load) and cold redundancy (standby), depending on the load subjected on the redundant elements [39]. Also, functional redundancy is often used in spaceflight, achieving functions with physically dissimilar mechanism not initially intended as a backup. An example for that is the use of on-orbit propulsion nozzles instead of a failed attitude nozzle. Besides elements that can be utilized as a backup, redundancy also

⁵² Contour was lost due to an improper installation of a heritage solid rocket motor that caused the spacecraft to overheat [69].

⁵³ An example for this is the Space Shuttle, where the development of the flight software was contracted to IBM, while the backup system was developed by Rockwell [188].

⁵⁴ In an independent assessment in 1988, several failures were detected in the backup flight software of the Space Shuttle [189]. One of the erroneous output was a sign error in the expression for the body flap deflection in the backup software on the General-Purpose Computer 5 [190]. The backup software, fortunately, was never engaged, though.

means that we can detect the failure and can transfer the function to the redundant component [13]. As an example of spaceflight redundancy, the reliability of a k -out-of- n redundancy can be calculated by [21]:

$$R = \sum_{k=m}^n \frac{n!}{k! \cdot (n-k)!} \cdot R_i^k \cdot (1 - R_i)^{(n-k)} \quad (23)$$

where k is the minimum number of elements needed and n is the total number of elements at the start of the mission [21]. Figure 2-36 depicts the reliability over time of selected k -out-of- n redundancies (left) and the effects of system redundancy vs. partitioned redundancy (right). Further redundancy schemes are depicted in Table 6-5 in Appendix B.

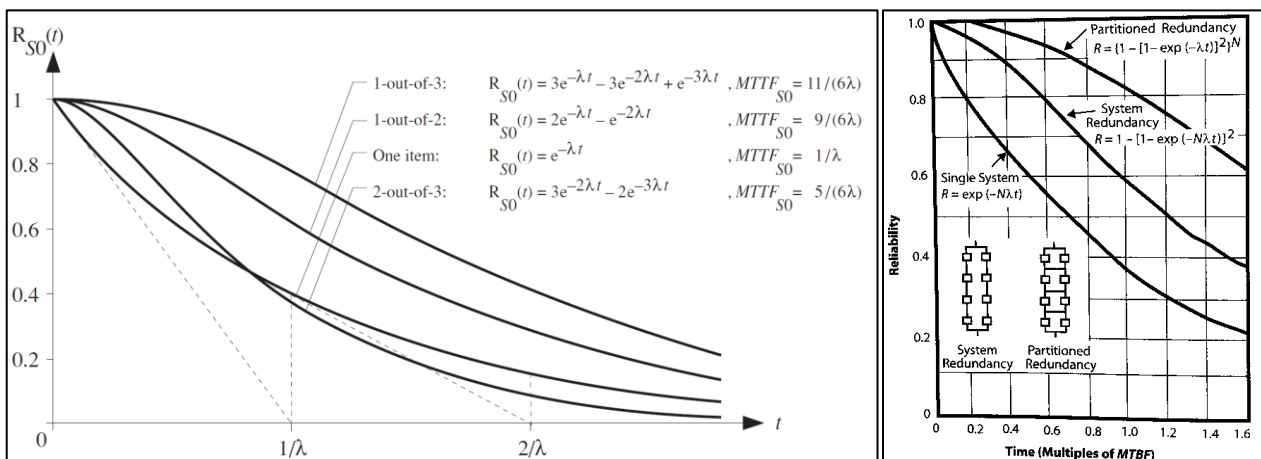


Figure 2-36: Reliability of selected k -out-of- n redundancies (left) and system vs. partitioned redundancy (right). Sources: left: [39], right:[21]

To conclude, redundancy is one of the oldest measures against system failures, and many spacecraft profited from planned or improvised backups in their mission. Approximately 17% of all failures in space can be coped with using workarounds, according to Hecht [13]. The Solar Terrestrial Relations Observatory (STEREO) program of NASA implemented both, redundancies in design, used for example in the power subsystem, but also on system level, launching two separate spacecraft to achieve the program goals [191]. While redundancy is common in traditional space missions, the limited envelope of CubeSats often forestalls this measure for those small missions. Replacement of a satellite with a spare, already used in some commercial applications, could be seen as alternative to stringent testing and reliance on flight-qualified components and/or systems [19]. Especially small satellites and CubeSats could utilize this approach, and improve their design stepwise. However, to do this, a basic functionality of the satellite must be ensured to get the required on-orbit feedback of the mission. Only then the knowledge can be used for successive satellites. And finally, when dealing with on-orbit spares and limited testing of spacecraft, the worsening space debris situation especially in LEO also has to be kept in mind (further discussed in Chapter 5).

The use of mass-production components is another important reliability assurance strategy mentioned by Hurley Jr. & Purdy [107]. We already learned that it is a misconception to see part level failures as the dominating source of space system failure. This misconception sometimes resulted in vast reliability requirements on part level or conservative design decisions, while neglecting other more prevalent reasons for mission failure. However, historically, it has been necessary to improve part level quality, not only for spaceflight applications but also for terrestrial ones. For example, the typical failure rates for a transistor used for the Mariner Mars 1964 mission were reported as 0.03 failures per 10^3 hours [77]. Nowadays, the typical failure rate of a space class transistor is estimated as 0.05 failure per 10^9 hours by the (also more than 20 years old) MIL-HDBK-217F [21]. As Sarsfield [28] noted, exhaustive efforts to improve quality and

reliability revolutionized the commercial electronics industry in the last decades. He presented the example of the Intel Corporation, which saw a sixteen fold increase of microprocessor complexity, fourfold increase in productivity while three-orders-of magnitude improving the quality of their products within 5 years [28]. The aircraft and automotive industries today rely heavily on mass-produced components, with the benefit of having typically completed their learning curve (i.e., reliability weaknesses have been removed) and being readily available on demand [107].

Today's automotive manufacturers build and test systems according to the Automotive Safety Integrity Level (ASIL) standard, in which for example ASIL D, the highest standard, represents random hardware failure targets lower than 10^{-8} hours, pushing part level failure rates also near 10^{-9} hours in the future. Assemblies, such as a classical DC/DC converter are built and tested to last 15 years, 900,000 power cycles, and more than 18,000 temperature cycles between -40°C and $+125^{\circ}\text{C}$ (accelerated testing) [192]. Thus, automotive electronics typically see a screening and testing process similar to electronics intended for spaceflight applications, except for vacuum⁵⁵ and high energy radiation – two topics we will discuss in Subsection 2.3.1 [26]. Table 6-6 in Appendix B depicts a comparison between typical tests of military, spaceflight and automotive electronics. In 1998 Hecht [13] reported failure rates of spaceflight screened parts (Class S parts) at about one quarter of that of military grade parts (Class B parts) and at one-tenth of that of high grade commercial parts. He later updated these ratios in the 2009 book chapter [21] to non-space qualified parts having failure rate 12 and 333 times that of space-qualified parts. Contrary to that, Sarsfield [28] remarked in his book that reliability of commercial applications has become extremely high, a factor that changed since the beginning of spaceflight, in which the performance requirements exceeded the capabilities of the electronic industry. Figure 2-37 shows the improvement in failure rate of electronic parts over time according to Sarsfield. Finally, Birolini showed in his 2014 book that commercial assemblies such as control cards for automatic processes (900 failures per 10^9 hours), multifunction telephone receiver (600 failures per 10^9 hours), or systems such as personal computers (9,000 failures per 10^9 hours) have achieved very low failure rates (all failure rates at 100% duty cycle).

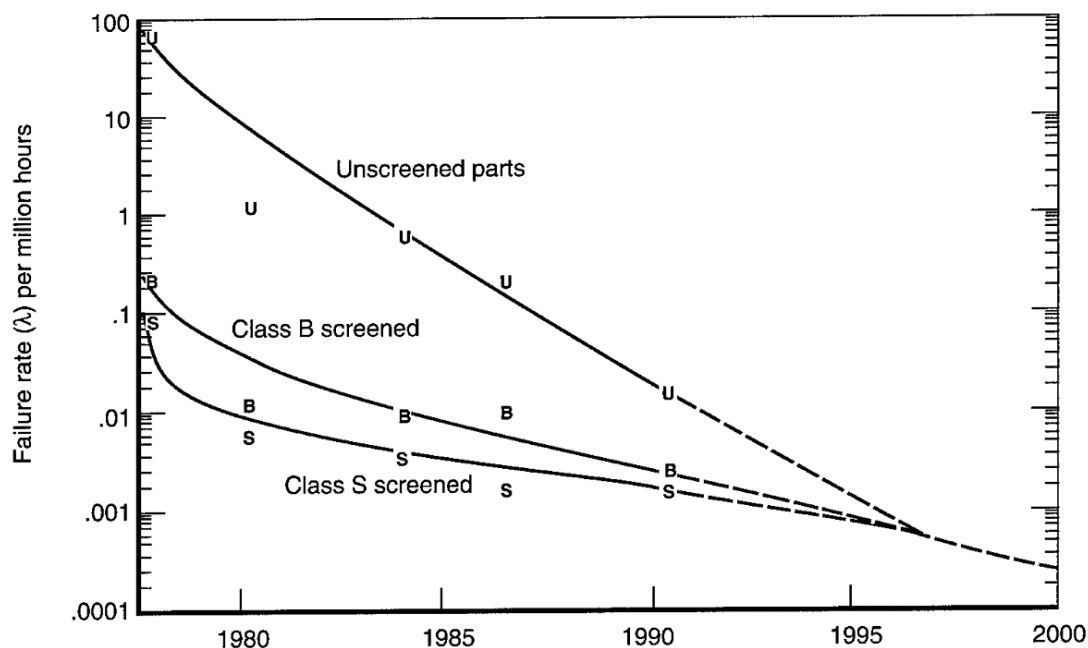


Figure 2-37: Historical failure rate of commercial unscreened, Class B (military) screened and Class S (spaceflight) screened parts. Source: [28]

⁵⁵ McDermott pointed out that vacuum testing is anyway assumed to be most valuable starting at board level [26].

Independently of whether there is a gap between the reliability of spaceflight screened parts and commercial parts nowadays, there are other factors to consider. First, as Molnau and Oliveri [193] pointed out, reliance on Class S level parts led to enormous costs of traditional spaceflight programs. Historically, the highest-grade parts available are chosen with the perception of reducing the risk of mission failure. The associated cost growth factor between commercial parts and Class S and Class B screened parts can be 10 to 100. More important, those parts can be purchased seldom in small numbers, since they stem from one production run, sometimes called batch or lot, which ideally will be purchased completely [27]. This is the second factor to consider while using screened parts. Fleeter presented the example of an integrated circuit, costing US\$50 as a commercial part and US\$500 in Class S screened version, but being only available in lots of 20 for Class S, increasing the cost to US\$10,000 [27]. As those lots are specifically screened and tested for spaceflight purposes, keeping a stockpile of all necessary parts is mandatory in traditional spaceflight, thus increasing cost [193]. Furthermore, this stockpile mentality slows down innovation, as more modern technologies with more elegant design solutions cannot be utilized when “old” proven parts are available. This does not mean that switching to the newest technology is the best option in any case, but a careful tradeoff has to be made between the reliance on screened parts, for applications in which it is necessary, and the utilization of COTS parts in other cases. Recently, obsolescence of older electronics became more and more a demanding topic for space agencies and the space industry, as some of their heritage electronic parts ran out of production years ago and their own stockpile is exhausted. Small satellites and CubeSats should not rely on screened components but rather profit from the huge progress of the commercial electronic industry in the last decades. Fleeter [27] mentioned a small satellite program in which US\$1.5 million were saved due to reliance on COTS instead of Class S screened components. Commercial parts can be procured in flexible amounts, and sometimes have lead times of 24 hours or less, compared with months for space-qualified devices [26].

Despite that, all reliability estimations associated with spaceflight and COTS parts and assemblies must be analyzed. As McDermott et al. [26] pointed out, mass-produced commercial components and assemblies benefit from the sheer number of items produced. 6-Sigma approaches, introduced by Motorola in 1985 to target virtually perfect execution of the production process, are standard in today’s commercial industry and result in 3.4 faults per Million items produced or a reliability of 99,9997% [193]. To guarantee a certain reliability of a component in a certain environment with a certain confidence level, a significant number of the same components has to be tested in that environment. To achieve a satellite reliability of 95% (single string, 1,000 parts) for example, we would need a part reliability of 99.995%. To demonstrate that part reliability on a 90% confidence, 46,052 items of each part have to be tested in a similar environment [26]. This is possible for the high numbers of parts produced in commercial industry, but not for spaceflight purposes. McDermott et al. continued their calculation with the RAD6000, a relatively popular spaceflight microprocessor. In total, assuming about 400 RAD6000 were delivered to spaceflight customers (a realistic number), and for each of them 10 were produced for testing purposes, the theoretical demonstrated confidence level of 99.995% reliability was 18%, assuming that none of the 4,000 microprocessors had a failure [26]. This does not mean that the RAD6000, nor other space proven parts are unreliable. It just shows the acceptance of the traditional space industry to rely on statistically (too) small sample sizes for spaceflight purposes. To conclude, reliance on specific parts produced for spaceflight can be a burden mission manager must take for some high-asset mission types. All others should have a critical look at the available options, and revisit Subsection 2.1.3 for the statistics of past missions. As soon as the parts or sub-assemblies are merged to a subsystem or system, the failure rate of the parts diminish often in the noise of the failure rate of a new design [26]. Failure rates of new designs will be dominated by systematic errors, as depicted in Figure 2-38, and this is the reason why thorough functional testing is needed for novel systems. As we have seen, the reliability of a spacecraft is more than just the sum of the reliability of its parts. Two remaining issues of COTS for space applications, radiation and vacuum, will be addressed in Subsection 2.3.1.

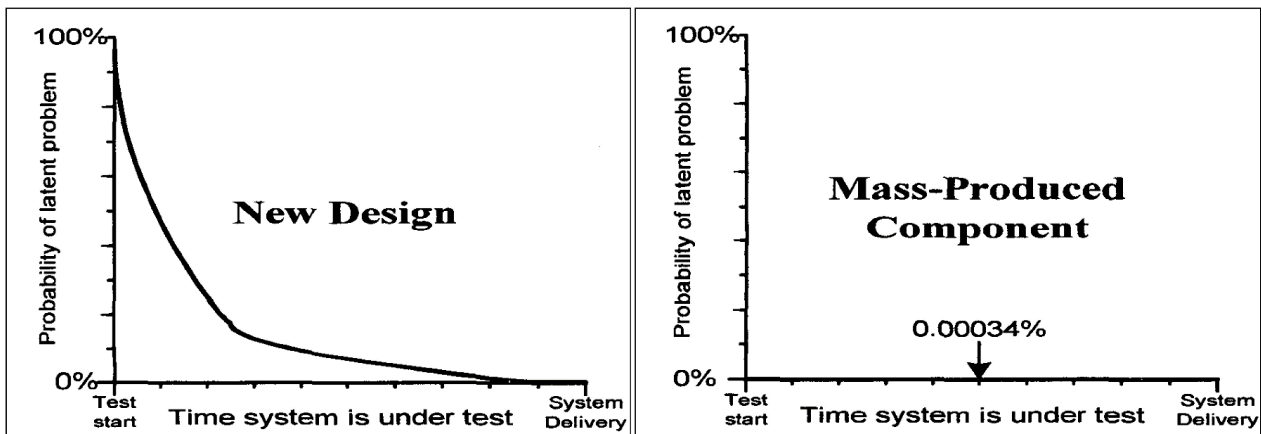


Figure 2-38: Difference between the probability of a latent problem in a new design over testing time (left), and the same probability on mass-produced components, put in place at a random time (right). The reliability of the new design is dominated by systematic errors to be removed by testing. Due to 6-Sigma approaches, the mass-produced component has an inherent failure rate of 3.4 faults per Million items produced. Adapted from: [26]

As the final reliability assurance strategy, we want to look into testing of spacecraft on the last pages of this subsection. As we have seen, testing is crucial for small satellites and CubeSats, since in many cases new designs are implemented and flown for the first time and redundancy is only partially available or not at all, due to their restricted envelope [28]. Furthermore, university-based teams often lack the necessary experience to make critical judgment calls during design and testing. This is especially delicate at the end of the development cycle, in which modest cost and schedule problems tend to have accumulated into larger problems [107]. Besides the multiple standards of spacecraft testing, there is also scattered knowledge about testing on past missions available. The findings will be summarized in the following.

The TLYF approach emerges as one of the key lessons learned from past missions. Many examples exist of mission failures attributed to not testing adequately in a TLYF configuration. As Ahmed [194] summarized, the TLYF approach means that no function, environment and stress must be experienced by a spacecraft (or its sub-assemblies) for the first time on-orbit. Although this might not be possible due to physical and engineering limitations in some cases, it should be enforced as often as possible [194]. As already pointed out, besides the mandatory environmental testing⁵⁶, functional testing shall be especially emphasized by small satellite and CubeSat teams, since many of them are using novel designs in hard- and/or software. White [195] defined TLYF as a systems engineering methodology to validate a system's ability to perform its mission, thus being more than only verifying requirements. Thus, TLYF can be seen as approach to partly unwrap the aforementioned tight coupling and complexity common in spacecraft, and detect faults and errors that emerge from the system level interaction and timing between subsystems, software, as well as the interaction of the spacecraft to the ground system. According to White [195] this shift away from requirement centric testing is necessary, because although many failed missions had met all stated requirements, they were never tested in a way that would demonstrate the successful accomplishment of mission objectives, and thus failed subsequently. TLYF has the goal of uncovering as many flaws as possible and not to prove that no flaws exist (which is impossible to prove). This does not mean to purposely break flight hardware by exposing it knowingly to damaging test configurations or environments. But if TLYF proves that such a failure exists while testing in a similar environment, this shall be viewed as a successful test, since otherwise the failure would have probably emerged on-orbit [195]. Thus, TLYF, in the full communication chain, should be one approach for small satellites and CubeSats to deal with design

⁵⁶ Besides proving that the system is able to withstand space and launch environment, certain environmental tests are mandatory for CubeSats to get accepted by the launch provider.

deficiencies and system-level interaction faults that otherwise would remain uncovered until on-orbit. TLYF and all other functional test approaches are then complementing, not substituting environmental testing.

FRACAS data of past missions is useful to shed light on the test and flight experience of spacecraft and reveal certain failure patterns. Usually FRACAS data are collected as failure reports on mission level, as done for example in the Near Earth Asteroid Rendezvous (NEAR) mission. The FRACAS of this mission revealed a pattern we have seen before. Most problems on-orbit were caused by design (128), followed by workmanship (37) and parts (9) and software anomalies (129) exceeded hardware anomalies (98) [36]. Aggregating such data over several missions would be even more useful.

This past lessons and patterns are key drivers for the improvement of the space system engineering process [196]. Databases exist at all larger space agencies as well as at private space entities. The Aerospace Corporation, for example, is collecting anomalies throughout development, launch and on-orbit operation in the so-called Space Systems Engineering Database (SSED), and furthermore internally disseminates lessons learned on past missions as bulletins [29], [128]. This has not yet been achieved in the small satellite or CubeSat domain, and moreover much of the data aggregated by the traditional entities is not publicly available. Thus, this class of satellites would benefit strongly from a centralized anomaly database [51] and more extensive data sharing through it.

The main database in Europe is the so-called Model- And Test Effectiveness Database (MATED), in which ongoing and concluded European space projects, their test and on-orbit anomalies as well as lesson learned are shared [197]. All information in this thesis was accessed through publications of MATED, and not via the system itself, since it seems not to be publicly accessible (anymore). Similar efforts exist in the US [64] and Japan [198], and the most important findings on all three are summarized in the following, mainly focusing on MATED.

In 2001, Tosney [199] reported a study on 47 space vehicles that showed the value of thermal-vacuum (TV) testing, as an additional amount of 1.9 failures per spacecraft were detected during TV testing after already 2.4 failures per vehicle were found in thermal cycling (TC) tests. He concluded that many TV failures required the unique stress of this environment in order to be found [199]. Arnheim [64] found design defects and workmanship errors to be the dominant sources of failures while studying 22 space vehicles. He also reported that recent spacecraft had more design defects than spacecraft from pre-1995, and attributed that to the increasing complexity of new missions [64]. In the same year, Messidoro et al. [200] showed the effectiveness of certain tests on system level, based on MATED data. They found functional and performance tests (~45% of all anomalies), integration tests (~33% of all anomalies), TV tests (~9% of all anomalies) and TC tests (~5% of all anomalies) to be most efficient in uncovering flaws. Interestingly, critical anomalies of the analyzed group were predominately caused by software (~75%), followed by electrical issues (~20%).

On system level, 44% of anomalies were found on the EM typically 6 months before launch, and 56% on the FM or proto-flight model (PFM) typically 3 months before launch. Finally, they reported that on-orbit flight data proved an infant mortality period of 120 days, similar to US data [200]. Maggiore et al. [201] updated these findings in 2008 and presented correlations between anomalies found in ground testing and on-orbit anomalies, normalized either over the number of spacecraft parts or a complexity index. In both cases, they found a decreasing rate of on-orbit anomalies with increasing number of ground anomalies detected beforehand [201]. Figure 2-39 depicts both curves. Although no further explanation is given on the correlation, it can be speculated that more anomalies being uncovered on-ground lead to less on-orbit anomalies.

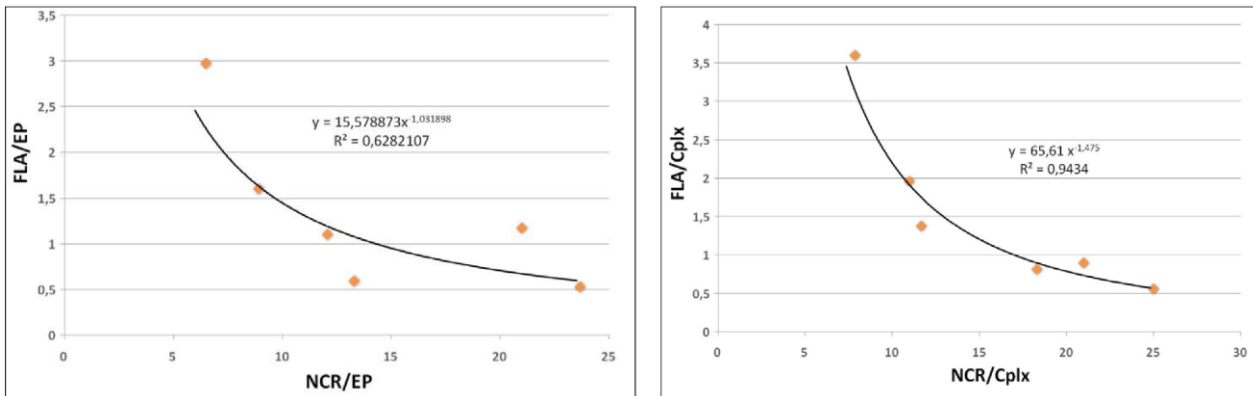


Figure 2-39: Correlation of anomalies in ground testing (NCR) to flight anomalies (FLA), normalized to the number of electronic parts (EP) of the system (left) and a complexity index (Cplx) of the spacecraft (right). Adapted from: [201]

In 2011 Brunner [202] reported that in the case of a series of spacecraft, anomalies at system level decline with increasing number of spacecraft and that design issues are reduced to almost zero at the fourth spacecraft of a series. He thus recommended to focus on workmanship, parts, and material effects while testing subsequent spacecraft. In the same year, Messidoro et al. [203] showed a pre-flight estimation of the early anomaly rate of the Gravity field and steady-state Ocean Circulation Explorer (GOCE) spacecraft. Based on MATED, they estimated an early anomaly rate of 7.3 per 10⁵ electronic parts for the first 120 days on-orbit (corresponding to six anomalies on the whole spacecraft) and confirmed this estimation with on-orbit data of 7 anomalies in the first 120 days [203].

In more recent studies, Laine et al. [204] and Pasquinelli et al. [205] again found design issues do be the most common cause for on-orbit anomalies (40% in the first and 50% in second study). Niwa, Takahashi & Shi [198] confirmed the infant mortality values from ESA in a Japan Aerospace Exploration Agency (JAXA) study, reporting an aggregation of approximately 60% of all failures in the first 120 days on-orbit. In one of the most recent updates on MATED, Brunner, Boerngen & Messidoro [206] showed the causes of all critical and major anomalies found during system level functional testing (depicted in Figure 2-40). Clearly, software faults, as discussed before, play a dominant role in today's spacecraft.

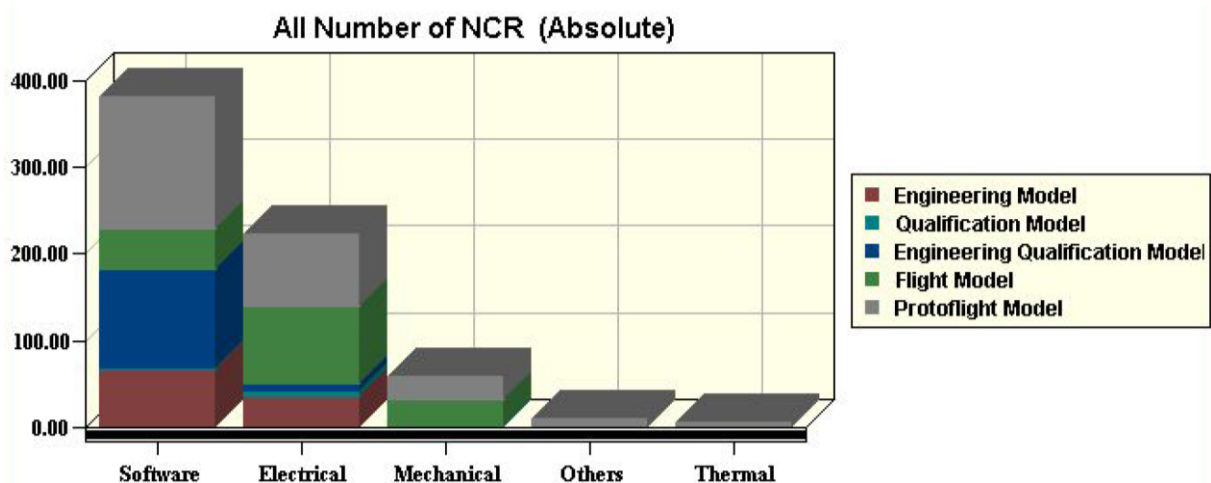


Figure 2-40: Cause of major and critical anomalies on spacecraft during functional tests. Source: [206]

Finally, in their last update in 2017, Brunner and Hagelschuer [207] confirmed the before stated effectiveness of functional tests (~45% of all anomalies detected), integration tests (~36% of all anomalies detected) and TV tests (~7% of all anomalies detected). The same pattern was observed when looking only

at critical and major anomalies⁵⁷. Also, the infant mortality theory was confirmed at $t = 120$ days. Most important, while studying 448 flight anomalies, they presented ground tests that could possibly have prevented those anomalies. Overwhelmingly, functional testing was found to be the best test method to prevent the studied anomalies, indifferent from the anomaly class, followed by an unknown category and TV testing [207].

To conclude this subsection on reliability prediction, reliability assessment and reliability assurance, we have seen that a significant number of on-orbit anomalies (design/engineering errors and software) is not covered in today's reliability prediction models, although test data and on-orbit feedback show the significance of these failure modes. Reliability assurance in traditional missions can be a costly undertaking, and some missions ended up at a reliability of zero due to project termination attributed to massive cost and schedule overrun. In 2007, the top 10 DoD space programs were overrun by over US\$32 billion, according to Wertz [30]. Although schedule and cost growth has been a historical part of spacecraft development programs [28], today's market as well as the public opinion will not accept overruns at these rates. Although maximized durability has been the important concept in spacecraft development, obsolescence slowly rises to claim this position [208]. Traditionally, satellites often took 10 years to be developed, and were built to serve 15 years or more, but in today's market they might end up being obsolete after 5 years (both from space but also from terrestrial competitors). According to a study from Saleh, Torres-Padilla, Hastings & Newman [209] obsolescence would reduce their optimal design lifetime to 3.5 years, also shifting from satellite operators back to satellite manufacturers.

The Cost of Quality, defined as the cost of all efforts to react to actual failures both before and after delivery, and to prevent failures from occurring in the first place, ranges between 30% and 50% during the phases of a typical spacecraft development program [28]. In February 2010, the GAO found that nine out of 10 NASA projects being in their implementation phase (out of 19 assessed) experienced cost growth ranging from 8% to 68%, and launch delays of 8 to 33 months. The 10 projects in the implementation phase produced a total cost overrun of over US\$1.2 billion [210]. All of this must be put into perspective to the risk of launch failures, which have not been accounted in this work so far. According to Tomei & Chang [211] the overall reliability of launches was 91.6% in August 2009, meaning that apart from all efforts to design and build reliable spacecraft, there is a 8.4% chance of failure in any case.

The first spacecraft launched into space were small and built around simple, inexpensive designs. Complexity increased in the following decades, and in some cases, this complexity was needed and justified to answer complicated scientific questions or to safely conduct manned missions. However, obsolescence, cost overruns, and delays pose a risk for today's satellite manufacturers to lose some of their markets to faster competitors or terrestrial applications that operate with shorter innovation cycles. The "rediscovery" of small satellites and the emergence of CubeSats could be a key to accelerate innovation cycles of space applications again. Thus, in the final section of this first chapter, we will summarize the most important characteristics associated with the miniaturization of spacecraft.

⁵⁷ ~48% functional, ~34% integration, ~8% TV [207].

2.3 CubeSats and the ongoing Miniaturization of Satellites

In this section we are going to summarize the ongoing trend of miniaturization of space hardware, focusing on CubeSats as the most important example for miniaturization and standardization. First, we will have a brief look into the historical evolution of spacecraft sizes, CubeSats and the role of COTS components in space applications. In the second section, we will see the most important characteristics regarding risk and reliability of CubeSats and conclude with examples of risk and reliability analyses of small satellites and CubeSats.

2.3.1 The ongoing Miniaturization of Satellites and its Implications

Originally, spacecraft started small, with Sputnik-1's mass being below 84 kg and Vanguard-1 weighing about 14 kg⁵⁸ [213]. Although many of the early satellites were small, there was no official classification of small satellites until 1992, when the University of Surrey's Centre of Satellite Engineering Research coined the terms "Microsatellites" (mass between 10 kg and 100 kg) and "Nanosatellites" (mass between 1 kg and 10 kg). This was later expanded by Janson [212], and the current definition has Picosatellites (mass between 0.1 kg and 1 kg) and Femtosatellites (mass between 0.01 kg and 0.1 kg) at its lower end. The upper limit of what is perceived as a small satellite changed also over time, and the definitions ranged up to a maximum mass of 400 kg or 500 kg, with the class between 100 kg and the upper limit sometimes being called Minisatellites [28] [214]. As of 2008, more than 860 Microsatellites, 680 Nanosatellites, and 38 Picosatellites have been launched. Figure 2-41 (left) shows the masses of the first 30 successfully launched spacecraft and Figure 2-41 (right) the masses of all Explorer spacecraft launched until 2005 [212].

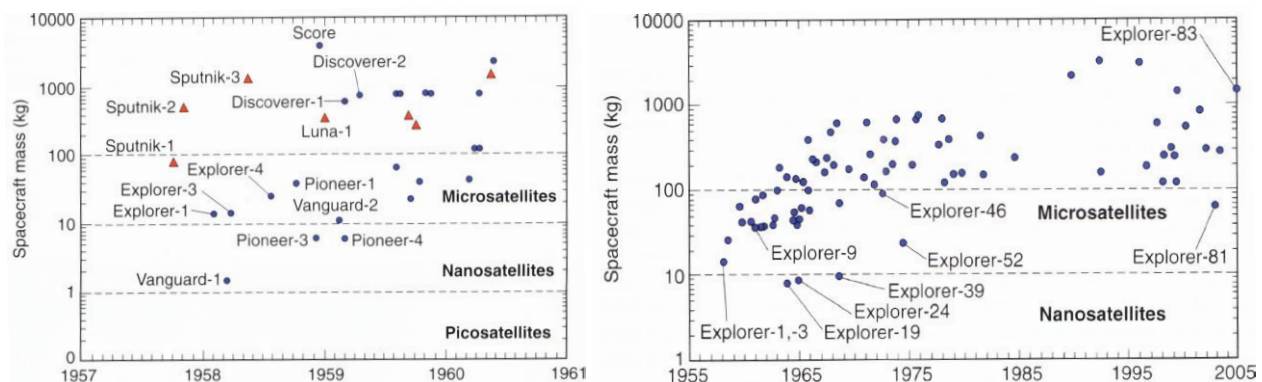


Figure 2-41: Launch mass of the first 30 spacecraft with successful launches (left) and the masses of all Explorer spacecraft launched until 2005 (right) [212].

The classification of satellites in terms of mass, despite being useful for launch and as a measure of complexity, was partly made obsolete by CubeSats, as we will see later. Also, the term "small" can be perceived differently. Huge differences in resources exist between for example a university program and NASA, in which for NASA "small" can mean a Discovery-class mission, and in between US\$50 million to US\$250 million of cost associated to it [214]. The first weather satellite (Tiros 1: 152 kg), the first commercial communication satellite (Early Bird: 39 kg) and the first amateur radio satellite (OSCAR I: 5kg) were all small satellites [27]. Over the years, the satellites grew with the launch capabilities and until about 1990, much of the conventional satellite technology focused on highly capable, but heavy and complex satellites [27]. As already pointed out, 1992 saw the introduction of the Faster-Better-Cheaper program at NASA and an

⁵⁸ Interestingly, the difference in mass between the first satellites of the USSR and the US can be attributed to the limited launch capability of the US at the beginning of the space age, resulting from the Eisenhower administration's desire to have an "open skies" policy for space, thus a civilian management of America's first satellite program. Vanguard-1 used a Viking research rocket as the first stage rather than one of the larger intercontinental ballistic missiles [212].

embracement of small spacecraft through the program by the then NASA administrator Goldin [214]. We already briefly saw the cultural difficulties that had to be overcome for Faster-Better-Cheaper in Section 2.1. The principle idea of Faster-Better-Cheaper rooted in the urge to cut the cost for spaceflight, similar to what was achieved in aviation 50 years after its invention. Although perceived as overall unsuccessful, nine of the first 10 missions of the program were a success, with Lunar Prospector and Mars Pathfinder (and Sojourner) being amongst those successful ones. However, in 1999, the success rate dropped and at the beginning of the year 2000 six of the overall 16 spacecraft of the Faster-Better-Cheaper program had failed. Looking more closely at those missions, three of the six failed missions were on a mission to explore Mars, which has always been a challenging target, and another one was a complex, cryogenically cooled telescope [14]. The clustering of failures in 1999 and the relative absence of failure before that suggest that the principle idea behind Faster-Better-Cheaper worked out, and that the “pick two” explanation is a rather unlikely one for the doomed missions [215].

Much research was put into finding an answer why the program failed. McCurdy explained the aggregation of failures by the Bearden rule, which states that complexity in missions increases cost and development time, with a linear relationship for schedule, and exponential for cost. Mismanagement, such as demanding too much complexity out of a limited budget and schedule then leads to failures [14]. In the unsuccessful Faster-Better-Cheaper missions often complex goals were planned for projects working with a fixed budget ceiling and a tight schedule, for some of their personnel for the first time. Automated spacecraft are complex and hard to build, as can be seen by the failures of the first six automated Ranger spacecraft [14]. An analysis by the Aerospace Corporation on 40 scientific and tech-demo spacecraft launched between 1990 and 2000 also showed a clear pattern. A no-fly zone, an area in which complexity is too high with respect to schedule and cost, emerged in their studies and is shown in Figure 2-42 [214]. This is confirmed by the successful missions of Faster-Better-Cheaper, such as the NEAR mission, in which the mission management simply only implemented changes with negligible or minimal disruption, and also incorporated only “half of the good ideas”, to limit complexity [215]. Faster-Better-Cheaper, if applied in a true sense, would have been a way to help mission managers reduce unnecessary complexity. As Ward [215] puts it, *“as long as we equate complexity with sophistication, complexity is going to eat our lunch, reducing our systems’ reliability and operational effectiveness”*⁵⁹.

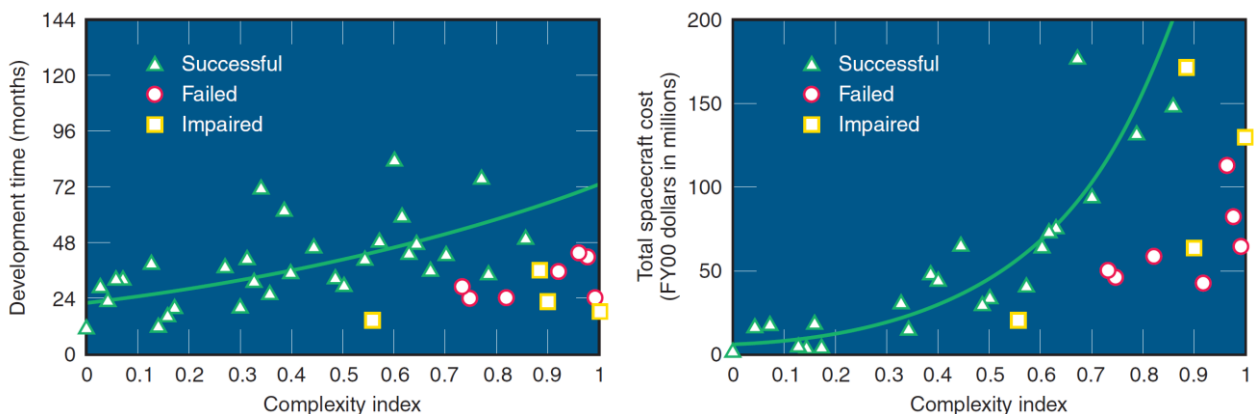


Figure 2-42: Successful, failed and impaired missions analyzed by the Aerospace Company. Source: [214]

McCurdy found inadequate techniques necessary to control teamwork in low-cost projects as a second reason for the mission losses of 1999. He reported that the budgetary constraints of the Faster-Better-Cheaper program led some mission managers to abandon their traditional, full scope system management approaches, yet they failed to replace them with approaches for low-cost programs [14].

⁵⁹ D. Ward, “Faster, Better, Cheaper Revisited: Program Management Lessons from NASA,” March-April 2010, Defense AT&L, page 52.

To summarize, although still perceived as a mistake by many in the space industry, the Faster-Better-Cheaper program showed that it is possible to reduce cost and time needed for spacecraft missions, but also to achieve reliable spacecraft while doing so. “Better” has to be understood as increase in relative, not absolute, capability, and complexity of spacecraft must always be seen as one critical driver of unreliability [14]. Originally planned as low cost and having reduced program-management, later missions of the Faster-Better-Cheaper program were ambitious endeavors with demanding requirements that saw complexity erasing most of their small satellite spirit [214]. Thus, it should rather be “Faster-Better-Cheaper OR Complex, pick one” than “Faster-Better-Cheaper, pick any two”. Overall, success-per-dollar is a better measure than success-per-attempt for spaceflight [14], [215] but for many space missions the “failure is not an option” attitude is still the main guideline.

Since the 1990’s, the total number of spacecraft and space programs has been growing, accelerating in the last decade, and so has the small satellite sector [28]. Besides of just having more spacecraft with a lower mass, this influences especially approaches to build, finance and manage risk for satellite systems. In many cases, today’s small satellites embrace higher risk tolerance and easier adaption of new technology, while operating with relatively low budget ceilings [213]. Their rise is fueled, similar to the rise of the CubeSats, by the ongoing miniaturization of electronics and an enormous market for commercial electronics that provides products with high reliability and high density at high production volumes. As Sweeting noted, Moore’s Law holds for several parameters of small satellite capability over the last two decades (depicted in Figure 2-43), and rapid response small satellites could change the economics of space [10]. Although we have seen the correlation between complexity and reliability, the relationship between spacecraft mass and reliability is not very well defined. On the one hand, increasing mass enables redundancy as a measure against failures. On the other hand, complexity mostly grows with mass⁶⁰, decreasing reliability again [15].

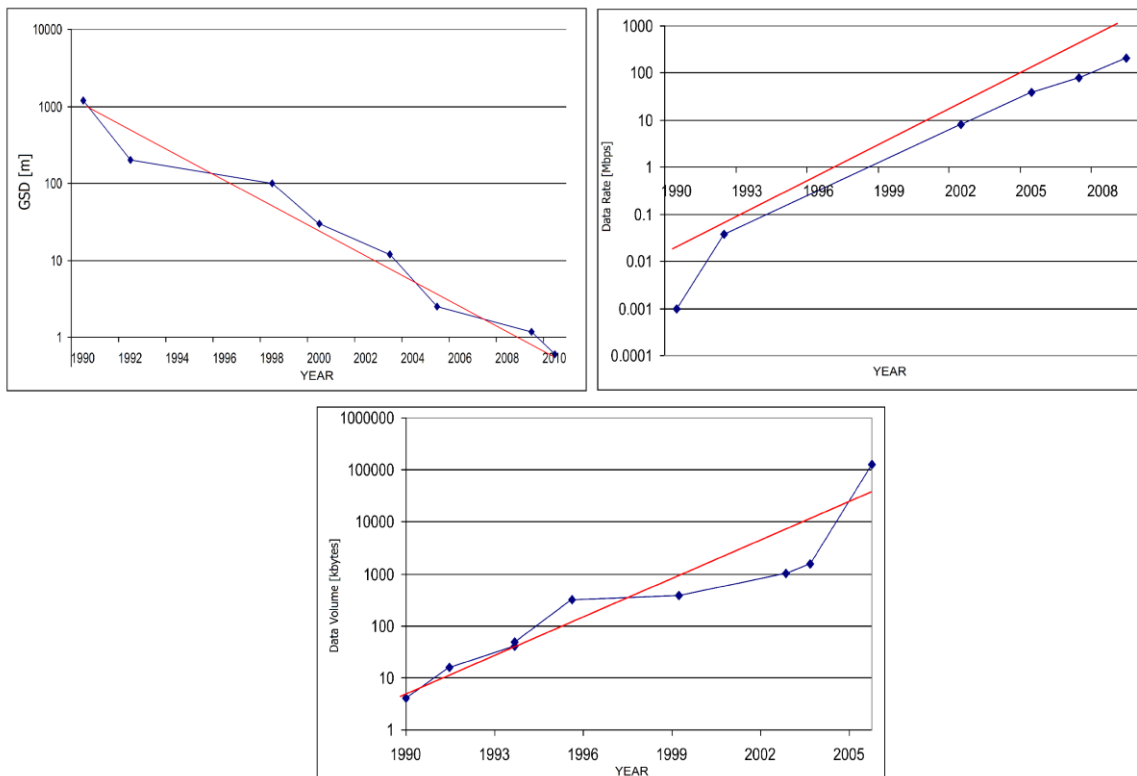


Figure 2-43: GSD (top left), Data Rate (top right) and Data Volume (bottom) increase of small satellites in a Moore’s Law-like rate over the last two decades. Adapted from: [10]

⁶⁰ As an important exception, software can cause a complexity increase without increasing the mass of the spacecraft.

Overall, small satellites should not be perceived as a replacement of larger satellites. Research questions that require complex instruments with billions of dollars of investment have to be answered with complex satellites that are built in a traditional, large way [30]. But exchange in terms of technology, lessons learned, and even human resources between small and traditional satellites, and already ongoing scaling efforts could help both spaceflight sectors [28]. As Horais [216] pointed out, the space industry lost much of its ability to conduct space research and development, because the risk of failure increased since the beginning of the space age. The reason for this is twofold: on the one hand, if we buy very complex, and thus expensive, systems we cannot afford spares, creating a high-risk situation, in which the system must work in any case. This often results in schedule and cost overruns, as for example project management sees the need to make Class A parts mandatory as a countermeasure against the increased risk [217]. The second reason is our own willingness to accept risk, as a society, and in general. As General Hyten [218], leader of the US Strategic Command put it: *“We’ve lost the ability to go fast, test, and fail. We tie the hands of our engineers and acquisition folk because we expect every test to work and if it doesn’t work it’s on the front page of the newspaper. We have got to get back to where we accept risk”*⁶¹. We already have seen in Subsections 2.2.2 and 2.2.3 that reliability tests of spacecraft have to be conducted with the target of uncovering as many failures as possible – otherwise systematic errors will remain hidden until launch. Also, Hyten noted: *“We went from zero to the moon really in about six or seven years. They went from the failure on the launchpad of Apollo 1 in January 1967 to walking on the moon in July of 1969, 30 months later. From the most horrible failure we had in the space program to the greatest success maybe mankind will ever have in space: walking on the moon for the first time. We were able to go fast”*⁶². In today’s global market, in which technology cycles are turning increasingly faster, the pace of technological evolution in spaceflight will determine if we can compete with terrestrial solutions. In many cases spaceflight is still an environment in which it can take 7-10 years from idea to a developed spacecraft, which itself will be in service for 12 years or more [217], yielding in the risk of on-orbit obsolescence [219]. We will have to progress from that. As Wertz [220] put it, we have to reinvent space, use modern technology and have the willingness to accept risk in order to accomplish more, faster, with fewer resources. Many of today’s missions are trapped in a spiral, as can be seen in Figure 2-44 (left), in which the growing cost of space missions leads to longer schedules and fewer missions, thus demanding higher reliability, leading to higher costs, longer schedules and fewer missions [30]. Small satellites and CubeSats can be seen as a way to reverse this cycle, as depicted in Figure 2-44 (right).



Figure 2-44: Traditional Space Spiral (left) and the potential of small satellites to reverse this spiral (right).
 Source: [221]

While considering all of this, we have to take into account that launching any spacecraft is a main driver of both, mission cost as well as unreliability, as we have already seen. The Space Spiral of Wertz can also be

⁶¹ P. Swarts, If America wants to succeed, it needs to learn to fail, top general says, SpaceNews.com. [Online] Available: <http://spacenews.com/if-america-wants-to-succeed-it-needs-to-learn-to-fail-top-general-says/>. Accessed: Feb. 27, 2018

⁶² P. Swarts, If America wants to succeed, it needs to learn to fail, top general says - SpaceNews.com. [Online] Available: <http://spacenews.com/if-america-wants-to-succeed-it-needs-to-learn-to-fail-top-general-says/>. Accessed: Feb. 27, 2018.

seen from a launcher point of view, creating a spiral of high launching costs, thus demanding high satellite reliability and vice-versa. Small launchers and series of small satellites or CubeSats, in which the emphasis is on the overall system reliability rather than on the individual satellite reliability, could be one way forward [19], [30]. As noted before, even the highest reliability in a design will lead to a reliability of zero in case of a termination of the project due to cost or schedule overrun, but also in case of a rocket failure and more generally, on every day the mission is late for the customer.

The rise of the CubeSats in the last decade was caused by many factors, while the standardization of these small vessels being the most important reason for their success. Similar to the success of the 40 foot shipping container, a standardized envelope for transportation was the enabler for easy and fast access to space [129]. For CubeSats, this standard is the CubeSat Design Specification (CDS) [222], which defines, besides others, the outer envelope, mass and the location of access ports of these satellites. The envelope itself, as aforementioned, is standardized into so-called units (U), measuring roughly 10 x 10 x 13 cm, with 1 U weighing up to 1.33 kg [222], [223]. What started as 1 U educational tools in 1999 [1], grew into multiple of units and missions with scientific and commercial goals. With over 300 Nano- and Microsatellites launched in 2017, and the majority of them being CubeSats, it is estimated that up to 2,600 of them will require a launch opportunity in the next five years [224]. While historically, as already pointed out, satellites are classified regarding their launch mass, CubeSats are making this classification superfluous. Figure 2-45 (left) gives an overview of current CubeSat sizes and masses. Today, a CubeSat can be anything between a 1U, below 1 kg educational satellite (old term: Picosatellite), over a 3U tech-demo mission with several kilograms (old term: Nanosatellite) to a scientific 27U, up-to-54 kg satellite⁶³ (old term: Microsatellite), as depicted in Figure 2-45. Although the launch mass is important in terms of complexity and costs associated with the mission, it is more important whether the satellite is standardized according to the CDS, thus having a clear, proven interface to the launch vehicle. This interface is the real key to the success of CubeSats, since the standardization of the CubeSats led to standardized deployment boxes (also depicted in Figure 2-45), which are now qualified to almost all current launch vehicles [225]. Thus, for the launch provider it is ensured that the CubeSat does not harm the mission in its role as auxiliary payload, and for the CubeSat developer, easy, cheap and fast launch access to a variety of launch vehicles guarantees that mission failure is not intolerable, and the Space Spiral rotates in the right direction.

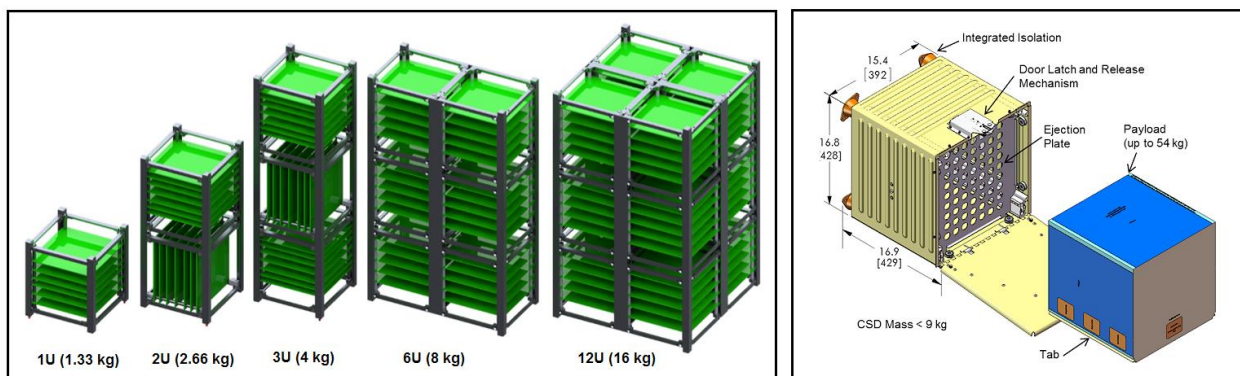


Figure 2-45: Standardized 1U CubeSat envelope and scaling of these units to larger CubeSats (left). Current biggest CubeSat size, 27U, and its standardized deployment mechanism (right). Sources: (left) adapted from: [226], (right): [227]

Today, CubeSats are valuable tools for project-based learning (PBL) education [228], where students are involved in hands-on projects, but also for innovation and commercial evolution of systems [229]. While traditional systems require many years to develop, CubeSats can reduce this development effort to faster cycles [225]. A 2016 NAS report [2] concluded that CubeSats show many characteristics of disruptive innovations, similar to the historical rise of personal computers, cellular phones or smartphones. Taking

⁶³ Note that, depending on the launch provider, more than 1.33 kg per U might be allowed for the mission.

root initially in simple applications at the bottom of the market, CubeSats have begun to move up, and the report found that the standardized form factor and the limited size are ingredients to this acceleration [2]. As aforementioned, the standardized deployer comes with the freedom to assume that a launch (at a well-known price) will be found when necessary, and in case of series of CubeSats, multiple launches could be used to “sequentially” develop the satellite, meaning that capability and complexity is added to later satellites of the series, and necessary experience gained with the first ones [225]. The US Company Planet, as others, uses this evolutionary approach for their CubeSats [43]. Of course, this means that the first satellite at least must successfully transmit data from orbit, as otherwise only very limited lessons can be learned. Due to their limited mass, size and resources CubeSats are mostly developed with COTS electronics, using very limited redundancy and more often limited testing in the program [225]. Typically, last stages of the system level testing are comprised of environmental testing, and this is often deferred until late in the project, when the satellite is fully integrated, and documents must be provided for the launch provider. Thus, design and engineering flaws (that could be detected on system level) are of big concern when studying the reliability of CubeSats.

Despite the higher risk involved, the NAS identified multiple areas in which CubeSats could be used to study high priority science goals [2]. The applications range from solar and space physics over earth science to biological and physical sciences in space. Overall, 25 science-focused CubeSats were planned from 2016 to 2018 by NASA and the US National Science Foundation (NSF). Beginning in 2011, the Radio Aurora Explorer (RAX)-2 CubeSat already proved the value of scientific CubeSats, delivering results about ionosphere turbulences caused by solar storms [230]. Some science goals will require the spatial or temporal coverage of constellations or swarms of 10 to 100 spacecraft, and CubeSats could be a suitable tool for that [2]. Swarms of CubeSats could form so-called “virtual satellites”, overcoming the size limitations of conventional optics [10], and then being used for synthetic aperture radar (SAR) applications [231] or applied for support of the Armed Forces [232]. The miniaturization of spacecraft is still underway, with centimeter-scale spacecraft, such as the so-called Sprites onboard the KickSat mission [233], already launched. Also PocketQubes, standardized satellites with a size of 5 x 5 x 5 cm and masses of around 100 gram have emerged, again as an idea of Bob Twiggs [234]. While this miniaturization will continue in the future, CubeSats are today the most common class of small satellites launched. As noted before, they go hand in hand with the incorporation of COTS electronics. However, even when using industrial or automotive qualified parts, the radiation and vacuum environment of outer space must be taken into account.

For CubeSat applications, a recent NASA study [235] found that 43% of all used parts are of industrial grade, 35% of automotive grade, 8% of military/space grade and only 3% of commercial grade⁶⁴. As reported in Subsection 2.2.3 and depicted in Table 6-6, automotive electronics typically undergo a screening process similar or beyond that of military or spaceflight grade components. For industrial grade electronics, this process is not as rigorous as for automotive grade ones, and commercial grade electronics are the least rigorously tested group of parts used in CubeSats. Furthermore, while automotive parts typically have similar operating temperatures (-40°C to +125°C) as military/space applications (-55°C to +125°C), this is slightly reduced for industrial grade parts (-40°C to +85°C) but heavily reduced for commercial grade parts (0°C to +70°C) [235]. Thus, automotive grade parts should be preferred over industrial grade ones for CubeSat applications, and it should be carefully considered if commercial grade parts are really inevitable in design solutions. A 1999 Applied Physics Laboratory (APL) study found COTS Plastic Encapsulated Microelectronics (PEMs) to be as reliable when subjected to HAST and temperature cycling (-55°C to 125°C) as their military/space grade, ceramic, hermetically sealed counterparts [36]. Temperature and temperature effects are tightly connected to the vacuum environment in space, and thermal runaway has to be prevented by ensuring enough conductive and radiative heat paths by design, and testing the device or assembly, mostly on subsystem and system level, in vacuum [26]. Only by testing

⁶⁴ The remaining 11% were of not clearly specified origin [235].

it can be guaranteed that COTS components will not suffer from the vacuum environment. This is also the case for the second problem to be addressed when operating in vacuum. Material degradation, stemming from outgassing, can deplete the electronic package and furthermore harmfully interfere with other parts of the spacecraft, such as optics. Also, for this problem, tests will either show that the COTS packages will outgas at an acceptable rate or lead to, if the part cannot be changed, the need for conformal coating, for example in a urethane derivative, to prevent outgassing. Summarizing, vacuum is a unique environment for which COTS components must be tested and which has to be kept in mind while designing the satellite and its Printed Circuit Boards (PCBs). But it is a systemic issue, and a microprocessor, if designed with enough heat transfer paths for space use, will not recognize the difference between operating in an automotive environment at 80°C or in a spacecraft at the same temperature [26]. This is of course a simplification, and for mechanical assemblies, especially moving ones, other characteristics have to be considered when dealing with the vacuum environment and COTS parts.

The remaining issue while using COTS parts for CubeSats is radiation, and since it is a broad topic, it can only be briefly summarized in this work. Generally, almost all COTS components can inherently tolerate limited amounts of TID, as experience in short term LEO missions. A NASA study in 2011 [236] reported on TID effects of selected COTS CubeSat electronics and showed that standard microcontrollers as well as flash memories can handle moderate TID doses very well⁶⁵. Thus, depending on the mission type, mission managers can decide if the used electronics have a good chance of surviving the TID rates planned, or if, for example, additional volume at the outer shell of the CubeSat shall be used for increased shielding. Using higher density materials and full outer shells instead of the skeletonized, classical chassis, would be one solution to further minimize TID [237], but with the negative side effect of creating secondary particles and Bremsstrahlung while doing so. SEEs is the other class of problems associated with the high energy radiation in outer space, though it is one that has to be solved architecturally [26]. SELs and other hard errors have to be dealt with in the design of the spacecraft, putting latch-up protections and other well-known measures as safeguards against these phenomena in place. SEUs and other soft errors can be corrected to a certain degree by modern error corrections techniques, some of them already standard on terrestrial COTS memory, or by applying redundancy in the form of multiple images of critical software. Nevertheless, SEEs currently remain a source for failure which extent is currently not fully understood for CubeSats, since many CubeSats fail for other reasons earlier in life so far. However, in the future, with more demanding missions and hopefully less infant mortality and DOA, more data on the effects of TID and SEEs in CubeSats will become available, and thus also more strategies to fully prevent their effects. De-rating of COTS components, thus operating them at much lower limits than possible, could be one easy solution to cope with both, thermal-vacuum and radiation issues [28], [36], [39].

To conclude this subsection, the ongoing miniaturization of spacecraft creates new ideas and novel applications for spaceflight at a higher pace than in traditional spaceflight, but also more challenges, both related to technical but also to management issues. The professionalization of COTS components manufactured for automotive and industrial applications lead to characteristics of COTS parts similar to traditional spaceflight/military grade parts, except for vacuum and radiation environment. Successful missions such as the Sojourner micro-rover, using modifications of a commercial Motorola radio modem of about US\$700, a standard Intel 80C85 processor and six small wheel motors of US\$100 each [14] showed that COTS can be applied even to demanding missions, if the environment and the mission time is carefully considered. For CubeSats, COTS is even applied on subsystem and system level nowadays, and these hardware can be purchased on short notice from many vendors, mostly having already flight heritage [213]. In all CubeSat missions, restricted envelopes decrease the possibility for redundancy, and tight schedules and resources pose the risk of negligence of important system level tests, as we will see in the next subsection on dealing with risks and reliability.

⁶⁵ Atmel ATmega microcontroller: 18.3 krad, Atmel Flash Memory: 15 krad (write) [236].

2.3.2 Risks and Reliability of CubeSat Missions

While small satellite programs have a natural tendency to be riskier than their traditional, larger counterparts, CubeSat programs, especially involving students, induce new risks that usually not occur in larger missions. As already pointed out, Class D missions are perceived as the riskiest missions by NASA, typically with low-to-moderate risk tolerance and complexity and mission life times between weeks up to one year [238]. Figure 2-46 shows the typical risks involved for all classes of missions, with Class D in red color. Note again that Class D typically has an upper limit of US\$250 million and ranges down to the University-Class Explorer (UNEX) Class D of about US\$15 million [238]. According to Tosney [199], high risk programs have a catastrophic failure rate of 25-30% in their first year, on average, while low risk programs have around 3-5% in the same timeframe. Putting that into perspective of the even lower resources and experience of most CubeSat teams, it is reasonable to categorize the risk of CubeSats as beyond Class D missions. Besides all of the challenges depicted in Figure 2-46, CubeSat teams often face an even higher turnover rate of their team, and students must often split their workforce between the project, classes, homework, learning and other work [239]. This workforce discontinuity, combined with frequent handovers, some of them unscheduled, was observed in many teams and special attention has to be given to this topic [240]. Furthermore, most university satellite projects are established within one dedicated faculty, although the co-location of the different disciplines involved would be highly desirable in any CubeSat project [241]. In a university environment, CubeSats face both higher schedule and cost risk, with tight budgets, long-duration funding cycles, and internal and external delay factors ever looming in many university-based programs. Also, miscommitment of students, and inexperience of most personnel involved are two important factors different from regular space projects [240]. Since small satellites and CubeSats are currently always secondary payloads on launches, lengthy launch delays are another source of failures, both on the technical side (loss of battery potency, corrosion, etc.) as well as on the management side (knowledge drain, especially in a university environment) [28]. Being already in a very restricted envelope, loss of design margin is also another risk common in many of these missions. Exhaustion of the involved personnel is another one, and both show the diversity of risks associated with CubeSat programs [28].

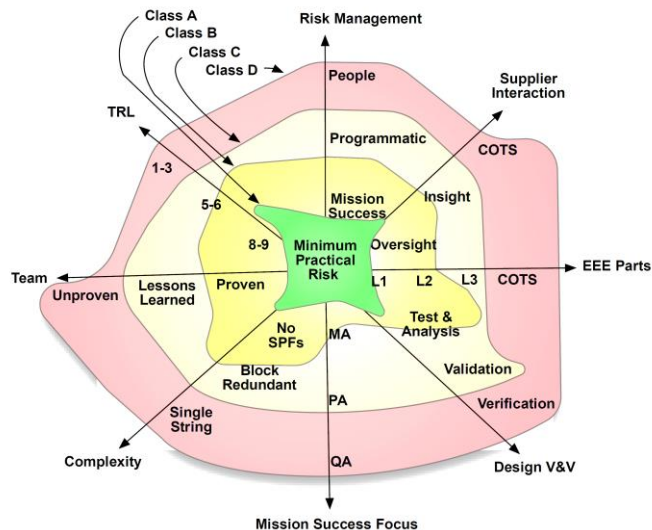


Figure 2-46: Risk surface of different classes of NASA missions. Class D, in red, has the highest risk. Adapted from: [242].

Although work by Brumbaugh [243] and Straub [228] presented ways to manage risk in university CubeSat projects, their budgets, schedules, and personal limits, the discontinuities in many factors involved inherently make university based CubeSat projects very risky endeavors. Thus, based on the lessons learned [244] of the first CubeSat of LRT, First-MOVE, and on the work of Brumbaugh [243], we created a risk assessment sheet for CubeSat projects, featuring both project management risks and technical issues as well as subsequent strategies how to mitigate them [245]. Although various mitigation strategies have

been put in place while we developed our current CubeSat, risks such as software design delays, missing documentation and limited time frame make our mission still highly risky [245]. We will gain some further insights on this in Section 4.3.

As presented in Section 1.2, the reliability of CubeSats is dominated by infant mortality and DOA cases, and many of them do not survive the first 6 months. For small satellites, due to their single-string design and reliance on COTS components, the prediction models would also indicate that many of them are highly unlikely to survive more than a couple of years. However, missions such as the Earth Observing (EO)-1 satellite of NASA [107], WindSat [107], ALEXIS [27] and data of small satellites built by Surrey Satellite Technology LTD (SSTL) [107] proved that if the small satellite survives the first year on-orbit, it is likely to survive the next five to 10 years. SSTLs data on 20 satellites launched between 1981 and 2003 show a measured average MTTF on-orbit of 6.4 years, while the prediction average was only 2.1 years. The reason given by SSTL for this fact is that the company focuses on improving reliability, not on the quantification of it [107]. This, combined with the experience and the heritage, minimizes the number of engineering failures remaining in a particular mission, thus maximizing the MTTF. According to the Aerospace Corporation [242], Class D missions do not require many standard procedures such as system level models, reliability growth trending, Probabilistic Risk Assessment (PRA), System FTA and FMECA, and reliability testing can be limited to safety critical items only [242]. Thus, it mostly depends on the spacecraft manufacturer what kind of reliability assessment safeguards are implemented to assure the extinction of engineering failures. Furthermore, according to the same report, anomaly reporting, failure analyses, and corrective actions are recommended, but not required, and might be captured informal. Also it is mentioned that typically, only a few key milestone reviews are held with few internal personnel additional to the associated engineers, but those reviews do not require external experts [242]. In CubeSat projects, this situation is usually worse, since the involved personnel usually does not have the necessary experience to conduct thorough reviews, and thus in many cases the review process is conducted on a very limited scale or completely neglected. CubeSats can achieve reliability over their simplicity, but as we have discussed before, simplicity counteracts redundancy in many cases. This is especially true for university missions, although current research tries to overcome these restrictions [246]. And furthermore, as noted before, adding redundancy always increases complexity, and examples such as the first Wakeshield mission [28] show that this can again decrease reliability, if not accounted for by increased resources and time. In some cases, CubeSats are perceived as assets of prestige for the stakeholders involved, leading to a “failure is not an option” mentality, which is clearly the wrong mindset when conducting such a project. As noted before, due to the costs involved, and justified by the simplicity of their designs and the short time frame needed for repairs, system level environmental testing is deferred until very late in the project in many cases [225]. This approach, as we will see, critically impairs the early detection of erroneous system level interactions, engineering and manufacturing flaws. Finally, CubeSats are often seen as simple and easy-to-build spacecraft due to their restricted size but if expected performance, mission goals, and functionalities are increased, the complexity rises, and this prevents simple designs from being put in place, and demands thorough reliability assessment efforts on-ground to ensure on-orbit reliability.

Study of past CubeSat missions conducted by Swartwout [11] showed the prevailing problems of CubeSat developers. His aforementioned 2013 study [11] found that for a third of 48 failed CubeSats no contact was achieved after launch, and Swartwout argued that two third of all failures arise from mistakes in functional integration (i.e., the spacecraft were not operated in a flight-equivalent state before launch). He concluded that many university teams have the misconception that system-level design, component-level design, and component-level assembly/test are the three biggest obstacles to mission success, and thus perform system level test with the expectation that the satellite will work flawlessly the first time it is assembled [11]. This is not the case for both university built CubeSats but also for larger missions, especially if the spacecraft is a first of its kind, as we have seen before. Swartwout pointed out that functional tests on system level are one area of concern for university teams, but many teams lack the time needed since they have planned for system level environmental testing, but not for system level functional testing at the end

of their development. He also reported in his paper that only two of his 112 studied CubeSats failed due to mechanical issues, and at most a handful due to thermal or radiation issues [11]. Again, this is not restricted to CubeSats as we have seen, and can also be observed in larger satellites, both on-orbit as well as while testing. Swartwout updated his findings in 2014 [247] and 2015 [248], and reported a 40% failure rate amongst university built CubeSats in the 2015 paper. In 2016, he updated his research [249] and distinguished for the first time between different groups of small satellite developers. So-called Hobbyists are the least experienced of these groups, and have a low-cost, fast turnaround approach. Traditionalists have a long history of successfully building spacecraft, and thus established procedures but are high-cost and low risk tolerant, since they build CubeSats in the same way they would build larger ones. He then specifically looked at the Operationally Responsive Space (ORS)-3 mission, flown in 2013, in which amongst 28 CubeSats flown, almost all of the Hobbyist failed (11 out of 13) and almost all of the Traditionalists succeeded (14 out of 15). All of these 28 satellites had the same set of acceptance tests required, and the same set of mission readiness review was performed by the NASA/DoD [249]. It can thus only be speculated where the differences were between both groups of satellites, but the evidence strongly points into the direction of experience, heritage and functional testing of new designs. This pattern is also confirmed by a later paper of Swartwout in 2017 [250] and his online database [5], in which it was distinguished between Hobbyists, Industrialists and Crafters for all launched CubeSat missions. While Hobbyists are not experienced developers, Industrialist equals the Traditionalists group, and Crafters are in between those two groups, thus being experienced builders of small spacecraft. CubeSats of constellations are excluded from this analysis. Figure 2-47 depicts the current success rate of all launched CubeSats except constellations ($n = 390$) analyzed in Swartwout's database as well as the success rates of Hobbyists ($n = 182$), Crafters ($n = 169$) and Industrialists ($n = 39$). The combined chance for DOA and early failure are almost 40% for all analyzed CubeSats. A difference between Hobbyists and Crafters in terms of chance for DOA and early loss can clearly be seen, although the higher rate of Crafter CubeSats with unknown status and mission still in progress might slightly reduce this gap. For Industrials it has to be considered that only 10% of all analyzed CubeSats fell in this group and many of them still had their mission in progress, although the data look better than for the other groups [5].

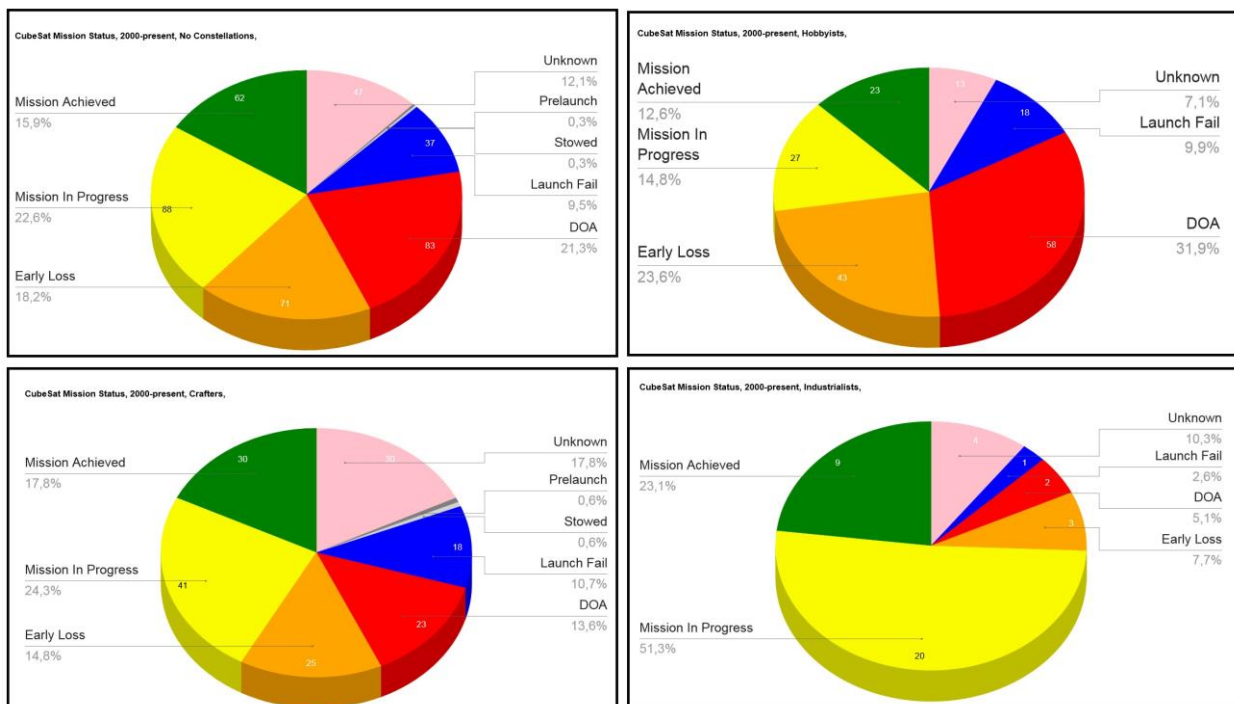


Figure 2-47: Success rate of all CubeSats launched (except constellations) up to 2018 (top left), of all Hobbyist CubeSats launched (top right), of all Crafter CubeSats launched (bottom left) and of all Industrial CubeSats launched (bottom right). Chart created on Tue Feb 27 2018 using data from M. Swartwout [5].

Finally, while studying university CubeSat missions in 2016 [251], Swartwout found correlations between the success rate and the experience of university built CubeSats. For university-built missions, the DOA rate was 25.7%, while 13.3% were lost early into the mission. 24.5% had some partial mission success while 21.2% claimed full mission success (the remaining portion are launch failures and unknown cases). He also pointed out that it took 40 years to fly the first 40 university class missions, and now there are 40 (or more) of those missions per year [251]. Considering the failure percentage that means that currently at least 10 per year could be DOA, and another five lost in early failures, if we remain at the status quo.

The final part of this subsection will deal with past work on how to estimate and assure reliability in CubeSat missions. In 2013, the ECSS specifically published the Product and Quality Assurance Requirements for In-Orbit Demonstration CubeSat Projects [252]. Being based on criteria for larger satellites, this standard unfortunately is very specific in some cases (e.g., no single point failure with critical or catastrophic consequences allowed) that are clearly not applicable for most CubeSats, in which redundancy is usually very limited. Also, no further advice is given on how to ensure reliable CubeSat hardware when developing it in restricted environments such as universities. In the same year, Fiala & Vobornik [253] used the MIL-HDBK-217F approach for reliability estimation of the embedded microcontroller system of the PilsenCUBE CubeSat. They reported a MTTF of 21.9 years and 13.4 years for different design options [253]. Also in 2013, Frazier, Rohrschneider & Verzuh [237] reported on CubeSat strategies for long-life missions. They emphasized that traditional methods of large satellites, such as random failure models and analysis to assure and predict reliability, are not applicable to CubeSats in which testing should be the preferred method for that goal. Furthermore, they also stated, that most orbital environments, except for radiation, are relatively benign compared to the environments most CubeSats parts usually would work in (e.g., automotive environment). They proposed qualification campaigns, based on HALT, to prove a three-year mission life in 4.2 months, using three additional satellite models that are not allocated for flight. Furthermore they would expose the flight model to an ambient +40°C for 4-6 weeks to ensure that infant mortality failures are pulled from the flight population [237]. Although they see cost reductions to traditional space hardware, their proposed solution seems to be too costly and too complex for most university teams.

Extensive work on CubeSat reliability was done by Obiols Rabasa [254], who published his dissertation on methods for dependability analysis of CubeSats in 2015. Studying the failures that occurred in past missions, he built a database containing failures of 175 CubeSats launched until December 2013 with data gathered from public sources. He excluded satellites where no information was available and reported 36 failures out of a group of 113 CubeSats, followed by a non-parametric and parametric analysis, depicted in Figure 2-48. He reported that the overall reliability of CubeSats drops to 69% after six months, while remaining at 60% after approximately 2 years, when the mission of most CubeSats ended anyways. The parametric model of Figure 2-48 is a two-parameter Weibull function with a shape factor of 0.2853 and a scale factor of 6,619.4 years. Although accounting for a reliability decrease of 14%, Obiols Rabasa excluded the DOA cases from his parametric model. Nevertheless, the studied group still showed clear infant mortality and he reported that his parametric model remains within 4 percent points of the non-parametric model [254]. This can be argued with since the Weibull plot in Figure 2-48 (bottom) shows deviations of more than 4% points early into the mission. Furthermore, DOA and infant mortality could both have the same origin, namely engineering faults, and thus should be accounted in the parametric models.

In the last section of his work, he proposed several methods to increase the reliability of CubeSats. As first method, activities of CubeSat developers were collected and presented. For this purpose, he sent out a survey but unfortunately got limited feedback (only around 15 developers replied [255]). Based on this limited feedback, recommendations and good practices, such as to follow the classical V-Approach, conduct thorough mechanical, thermal and radiation analysis, and a list of design best practices for every subsystem as well as recommendations to avoid of certain materials were reported. Furthermore, he observed a correlation between tests and analyses conducted and the later mission success as well as the level of redundancy implemented and mission success, and found some relation between satellites that

have been re-tested after failure and their failure rate on-orbit. He also suggested that FMECA is highly useful for CubeSat projects and reported on results of a FMECA for the e-st@r-I CubeSat [254].

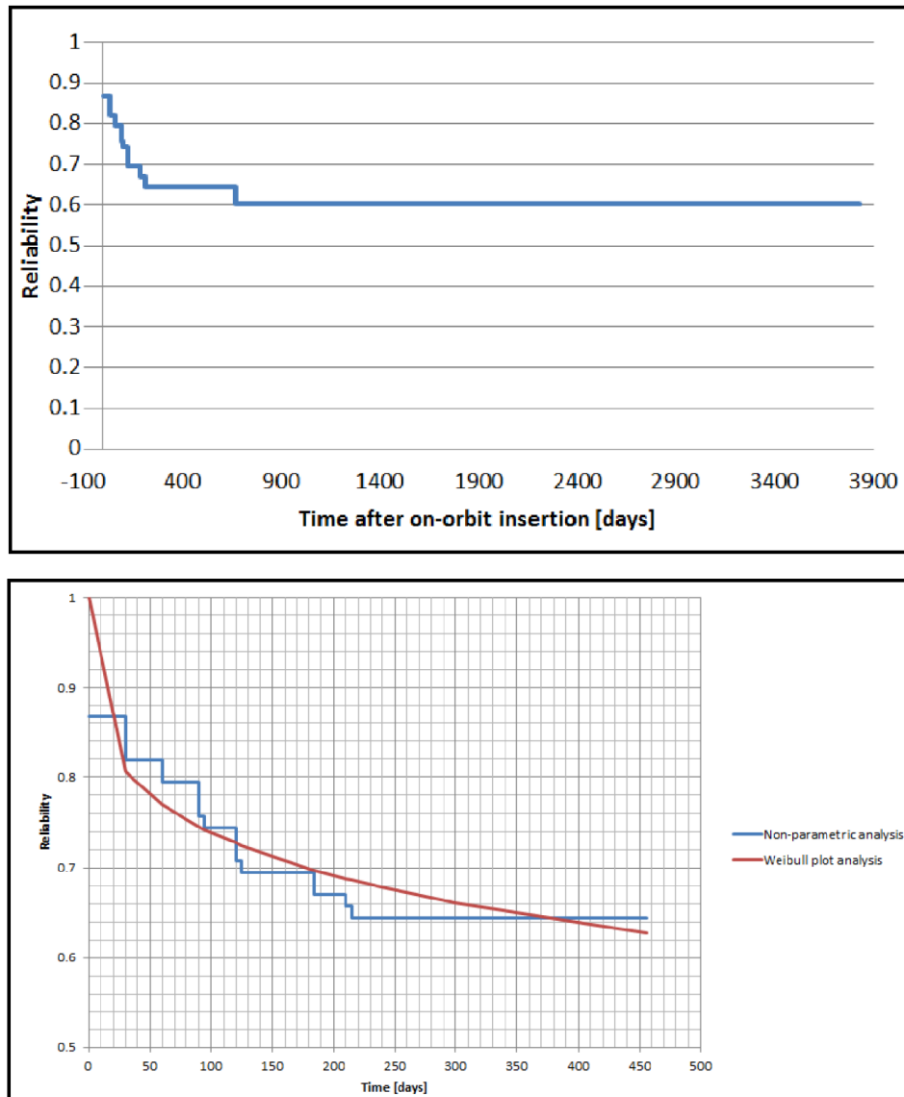


Figure 2-48: Non-Parametric reliability (top) and parametric Weibull model (bottom) of 113 CubeSats analyzed by Obiols Rabasa. Source: [254].

As a second method, Obiols Rabasa presented a tailored ECSS approach, similar to the one already mentioned in this subsection. Based on this tailored approach, he foresaw a traditional set of environmental tests and functional tests as the verification process for their new CubeSat e-st@r-II [254]. Although the proposed process is similar to traditional approaches in larger missions, functional testing on system level is not specifically tracked and the success criteria is mostly the fulfillment of all requirements. As we have seen before, this approach is not exhaustive, since, historically, failures sometimes occurred early in space missions despite all requirements having been met during testing. In the last part, he presented work on mission-oriented reliability, based on system level redundancies within a constellation or swarm of CubeSats. Based on the earlier presented parametric model, he showed that 12 CubeSats, with a minimum of nine operating, could achieve a higher reliability than a traditional satellite after 10 years of insertion into orbit [254]. This assumption must be taken with care, since his parametric models are excluding DOA cases and the reliability data beyond 2 years of lifetime becomes very scarce for CubeSats, making the prediction for 10 years lifetime statistically questionable. Nevertheless, the presented system level redundancy in swarms and constellations could be a promising way for future commercial and scientific missions, not only

for CubeSats but also for larger satellites. The threshold of DOA and infant mortality cases acceptable in such settings must be evaluated in the future.

Looking at small satellites, Cho [256] and later Cho & Faure [257] published approaches for reliability assessments on satellites. Using the earlier presented Duane growth model, Cho reported on the need to improve testing of small-scale satellites, since infant mortality is also still dominating in this class of satellites. He presented theoretical studies of different system level testing times and the resulting theoretical improvement of on-orbit reliability [256]. This process was later applied by Cho & Faure on the small satellite HORYU-IV [257]. They tracked the improving rate and reliability growth during assembly, integration, and testing and calculated a reliability of less than 30% 24 hours after launch, based on the growth experienced, collecting about 80 failures in 500 hours of testing [257]. These results were updated in a 2016 paper by Faure, Tanaka & Cho [258], reporting on approximately 2,000 hours of testing on the first EM, the second EM and the FM of HORYU-IV. In this paper they showed that most failures emerged early in the testing process, with more than half of the failures of the first EM discovered within the first 50h of testing and more than 2/3 of the failures of the second EM discovered within the first 150h of testing. Overall, they found about 160 failures in roughly 2,000 hours of testing [258]. In 2017, two updates were provided by Faure, Tanaka & Cho [259] [260] with a taxonomy of all failures and Duane growth models. Figure 2-49 shows the reliability improvement over time (top) and the resulting Duane plot (bottom). For the failure rate a shape parameter $\beta = 0.83$, and a scale parameter $\lambda = 0.57$ were estimated. Thus, the reliability growth parameter of the testing campaign would be 0.17 [259].

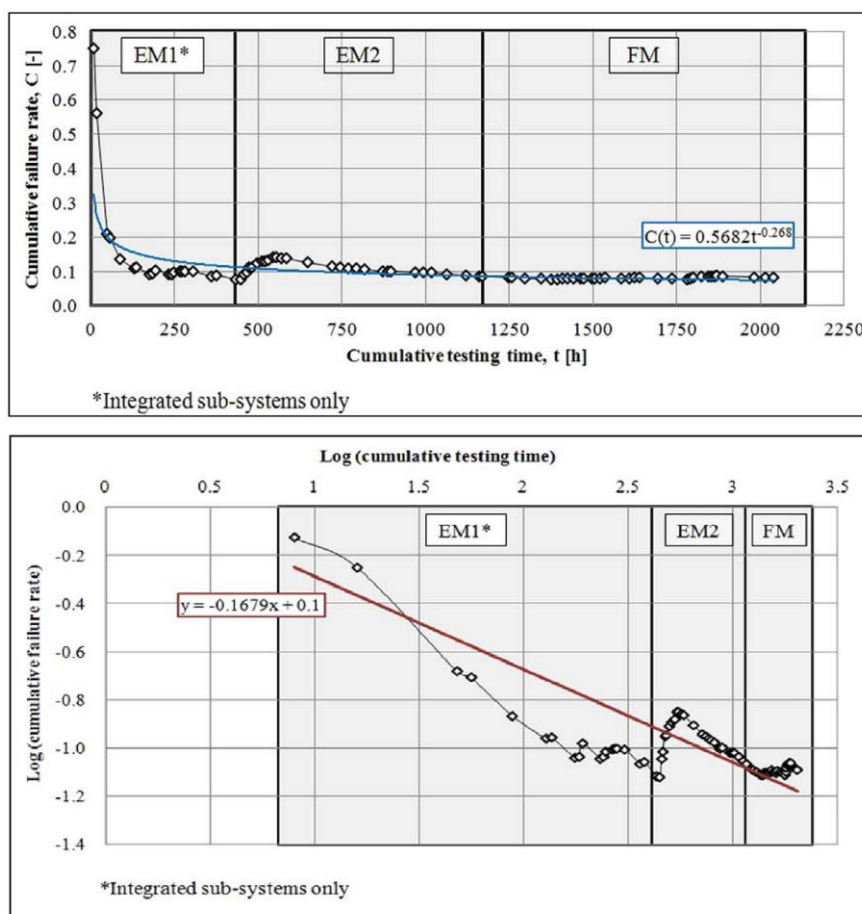


Figure 2-49: HORYU-IV cumulative failure rate (top) and resulting Duane plot (bottom). Source: [259]⁶⁶.

⁶⁶ Note of the author: It is assumed that the exponent of the cumulative failure rate function in Figure 2-49 (top) is -0.168 instead of the depicted -0.268 in the paper ($\beta = 0.83$).

Overall, Faure et al. reported that 73% of all failures were caused by electronics, structural elements, design and software [259]. While most design failures emerged while testing the first and second EM, the FM tests were dominated by failures caused by software (39%), followed by unknown causes (21%), workmanship (15%) and design of electronics (9%) [260].

Due to the increasing relevance of software for small satellites and CubeSats, we will conclude this subsection with a report by Jacklin [261] on the verification and validation techniques currently used for small satellite software development, published in 2015. He reported that small satellites usually do not receive the same level of software assurance as traditional satellites, as they typically have more restricted budgets, but the cost for verifying and validating software is usually not smaller. Although the V-shaped software lifecycle diagram would be a standard way of validating the system level functionality against the requirements, this process tends to be difficult for small satellite and CubeSat developers. Often, many of the initial software requirements are assumed by the software developers, in order to start developing software that can be tested and reviewed early in the process [261]. However, this leads to software problems that emerge late in the validation phase, which are traditionally hard to tackle for the developers due to restricted budgets and time. According to Jacklin [261], most small satellite developers use simulation and testing for software reliability assurance, but those simulations are often conducted with lesser fidelity and rigor than in their larger counterparts. As we have seen, many software problems emerge as system interaction problems. Thus, again, thorough system level functional testing combined with reliability assessment and growth models could be a way to prevent a too early termination of the test campaign or too little resources planned for adequate system level functional testing.

To conclude, the ongoing miniaturization of spacecraft has seen a significant advancement with the emergence of CubeSats, which are designated by some as a disruptive innovation of spaceflight [2]. More than 700 of these standardized satellites have been launched so far, and current projections see the launch rates further climbing. Today's CubeSats experience higher infant mortality and DOA rates than their larger counterparts. Many errors are not detected before launch due to lacking system level functional testing. As we have seen, reliability prediction is based on constant failure rates, and thus clearly not applicable for reducing these cases. Infant mortality, hence flaws in design, engineering and workmanship, can only be found by testing, thus assessing the system's reliability. Historical examples of small satellites working way past their projected orbital lifetime show us that there is no basic showstopper in producing more reliable CubeSats. Limited resources, experience, and time are factors almost every university CubeSat team has to face. Nevertheless, if satellite development is carefully planned around a system level reliability growth campaign, and thus system level interaction failures and engineering flaws are detected and corrected, the author of this thesis believes that the infant mortality and DOA rates for CubeSats can be reduced. Of course, such a campaign would be complementary to thorough environmental testing, and not substituting it. We will see a potential way of doing that within a real CubeSat development program in Section 4.3.

3 Gap Analysis & Objectives

“The only way of discovering the limits of the possible is to venture a little way past them into the impossible.”

– Clarke's Second Law, Arthur C. Clarke

“Engineering is done with numbers. Analysis without numbers is only an opinion.”

– Akin's 1st Law of Spacecraft Design

Traditionally, NASA missions spent an overwhelmingly large fraction of their development time in preliminary design and critical design, and only a minor fraction in manufacturing and assembly integration and testing. By doing that, the risk of failures is usually pushed back to later stages, and bought by many re-designs [43], often using margins and redundancies. For CubeSats, this traditional way of developing space hardware and pushing back risks is risky itself, since developers often lack the experience for critical judgment calls in the later test campaign and they cannot rely on big margins and redundancies in their missions. Statistical data show that CubeSats suffer from infant mortality, with the group of university-built CubeSats being an extreme case [251], [109]. As described in the last chapter, increasing reliability at component level should actually increase spacecraft reliability in parallel, and this would be true if random part failures were the dominant cause of spacecraft failures. But this is not the case, and thus also most classical methods of reliability estimation, due to their reliance on constant failure rate models, fail to produce useful data to prevent infant mortality.

Modern spacecraft, especially miniaturized ones, are more or less an assembly of flying microcontrollers, processors and sensors, heavily relying on software, and many of them combine tight coupling with complex interactions. Failure causes for infant mortality can range from design failures, software failures over operator errors to workmanship failures (but are not limited to that). Furthermore, as we have seen, many of the early failures will only emerge on system level, and some remain uncorrected until launch. In larger missions, this situation can often be resolved by delaying the launch, or ground intervention and redundancy if the satellite is already in space. Especially the latter two have greatly reduced the severity of many failures in past missions [13]. For CubeSats, all three options are limited. The tight envelope usually restricts the utilization of redundancy. Communication channels are often very limited, and the hardware, both on-ground as well as on-orbit is not as sophisticated as in larger missions. Launch delays cannot be mandated by CubeSat teams, as they are only a small fraction of the launch mass, but in many cases the launch could be switched by paying an extra fee. Overall, most CubeSat teams, especially university-based ones, lack the experience of planning the needed time and resources for adequately maturing their system before launch. As we have seen in the study of Swartwout [11], many of the CubeSats failed due to poor system-level functional testing, as the satellites were not operated (or not long enough operated) in a flight-equivalent state before launch. This is the biggest knowledge gap to be closed in order to achieve that CubeSats do not suffer from the same DOA and infant mortality rates as today. As Ogamba [106] pointed out, today's reliability assessment methods could help identifying critical reliability problems, but most of

the methods are not well suited for seamless integration with currently used systems engineering processes. While looking at historical data, past CubeSat missions, and the unique environment of university-based CubeSat development teams, system level functional testing and reliability growth modelling could be a way to reduce infant mortality.

The main goal of this thesis is to decrease the chance of facing DOA or infant mortality in MOVE-II down to an acceptable level⁶⁷. The experience of applying the reliability assessment methods approaches to test hard- and software as early as possible to MOVE-II will hopefully help to close the gap of infant mortality of today's CubeSats. Although it is the most complete source of CubeSat reliability data, the research by Swartwout shows only the success and failure rates of past CubeSat missions. The time dependence of both parameters remains unknown [12]. The work of Guo [109], although incorporating time dependency in his models, does not specifically focus on CubeSats as a class of satellites, and furthermore only has an incomplete set of CubeSats launched so far in the database. Thus, a minor knowledge gap to be closed is to collect reliability data of past CubeSat missions, and to create a CubeSat reliability database out of it. This effort begun in mid-2013 and will be presented in Section 4.2. Therefore, later reliability databases, for example of Obiols Rabasa [254] and Palla et al. [113] were not published when this work was done. Another minor knowledge gap to be closed is the lack of a FRACAS in most current CubeSat projects. As we have seen, a "learning curve" typically appears when manufacturing more than one satellite [87]. Thus, starting with the second satellite of a series, engineering problems and design deficiencies have typically already emerged and can be corrected afterwards. In companies and established entities, heritage and knowledge of past missions can partly replace this if building a first of its kind. For most CubeSat projects this is not the case, as the satellite is often a first of its kind, so little to no knowledge is available on design and engineering problems of the past. A method to prevent engineering and manufacturing problems from slipping through testing and emerging in space would be the use of a controlled FRACAS within CubeSat projects, also in order to capture knowledge of past problems for future projects. Data of the FRACAS could then also be used for the aforementioned growth modelling.

Finally, research gaps also exist with respect to larger missions, influencing smaller satellites as well. Reliability analyses of larger missions, for example done by Castet & Saleh [116], have to be studied to extract time-dependent failure behavior from their data as well. Some of their parametric models must be analyzed, as the shape and scale parameters used, and the conclusions drawn are arguable. The increasing pace of technology cycles, also for larger missions, will result in greater use of COTS parts [217] and a larger risk of on-orbit obsolescence. As we have seen, ESA [70] evaluated current reliability prediction and assessment models as insufficient, since large differences exist between predicted performances and the actual on-orbit reliability. This is mainly due to the high occurrence of systematic failures, thus non-component causes, such as design deficiencies, software flaws and manufacturing defects. ESA concluded that a key area for improvement would be "*A more holistic approach to reliability prediction accounting for non-parts related system failures causes*"⁶⁸. As we have seen in past chapters, software flaws and engineering mistakes can hardly be predicted by theoretical models – they must be assessed in systematic tests and could be projected in the future afterwards using models such as the aforementioned MATED platform. While doing this, any reliability projection should be handled with care, and in the opinion of the author of this thesis the very first step should be to mitigate early failures in CubeSats by using this approach. Only at a later stage, when more on-orbit data become available, and correlations can be found, reliability estimations should be conducted for CubeSats. To conclude, all research gaps and objectives presented also exist to some degree in larger missions – so besides the overall goal of improving the DOA and infant mortality rate of future CubeSats, methods and lessons learned of this process could also help some of the larger missions to succeed.

⁶⁷ Since CubeSats are inherent risky endeavours, this level cannot and should not be zero.

⁶⁸ L. Bianchi, "New Reliability Prediction Methodology Aimed at Space Applications: Briefing Meeting with Industry, ESA/ESTEC Product Assurance and Safety Department," Apr. 2016, page 9.

3.1 Objectives & Anti-Objectives

Based on the already presented research gaps, the primary and secondary objectives of this thesis are presented in the following. The main goal of this thesis is to reduce the chance of facing DOA or infant mortality in the MOVE-II CubeSat to an acceptable level. Since this work was implemented in a university-based CubeSat project, and such educational projects represent one extreme of satellite development, all results and conclusions presented have to be seen in the light of this. Nevertheless, although there is a gap between traditional satellites and small satellite paths, this gap is not a bottomless canyon, rather it is a continuum of solutions in between [213]. Thus, results, although found for a university environment, could also be helpful to some scale in larger, traditional missions, or anywhere in between these two.

Time-dependent failure behavior of past large-scale as well as CubeSat missions shall be researched as one objective in this thesis, studying the influences of infant mortality, constant failure rate and wear-out and scrutinizing the underlying reasons. To study past CubeSat missions, a reliability database is built and root causes for failure have to be analyzed. Finally, as Goel & Graves [40] noted, reliability assessment methodologies are needed while developing reliable products in today's global markets in which electronic products are rapidly changing. Such an assessment methodology shall be developed and applied to a real CubeSat to decrease the chance of DOA and infant mortality. Thus, the three primary objectives of this thesis are:

Primary Objectives

- 1) **Extraction of the time-dependent failure behavior of satellites from today's in-flight reliability data.**
- 2) **Collection of CubeSat in-flight reliability data and extraction of the time-dependent failure behavior.**
- 3) **Development of a reliability assessment method for CubeSats to identify, track, and subsequently solve possible DOA and infant mortality causes and thus significantly and efficiently increase the reliability of university-built CubeSats.**

Again, it is important to note the differences between reliability assessment and reliability predictions: while the former obtains data from tests or field data, the latter work with calculations on the basis of component failure rates [39]. Since all historical data suggest that CubeSats are dominated by infant mortality, caused by non-random faults, reliability assessments are the preferred option to increase their reliability. All of the proposed solutions are complementary to environmental tests, in which it has to be proven that the selected hardware can survive launch loads and work in the space environment. Besides the primary objectives, three secondary objectives were formulated for this thesis. The first two will help to support the primary goals, and the third will be helpful for design-tradeoffs in future missions, when hopefully the infant mortality and DOA cases will have decreased significantly.

Secondary Objectives

- 1) **Create a FRACAS that can be used in a university environment to prevent engineering and manufacturing problems from slipping through.**
- 2) **Improve the TLYF approach for CubeSats in order to generate accurate data for the system level reliability assessment.**
- 3) **Develop a reliability prediction method for CubeSats, to efficiently trade-off design options in early phases.**

It shall be noted that the abovementioned reliability assessment and TLYF approach should be done on system level. Nevertheless, reliability on subsystem-level will be ensured through a traditional review and verification process, and data from the system level test can be used to identify weak links in design and implementation on lower levels of the design. Besides the objectives of this thesis, it is also important to identify the anti-objectives, i.e., results that cannot be found in this thesis.

Anti-Objectives

This thesis is embedded in a university-based CubeSat project, so it is not the objective to draw any conclusions for larger satellites, commercial CubeSat and SmallSat missions or manned space systems. Also, all absolute numbers in terms of reliability must be taken with care as the principal goal of this thesis is to decrease infant mortality rates, not to produce the “best” or “most realistic” estimation of on-orbit reliability. Any conclusion regarding the root cause of a spacecraft failure must be carefully evaluated. This is even more the case for CubeSat failures, for which on-orbit data and information is rarer than in traditional missions. Thus, many of the failures in the database are the most probable reason for failure after a thorough root cause analysis by the developers, rather than a 100% guaranteed fact. The objective of this work is therefore not to prove irrefutable root causes for all CubeSat failures but to present the time-dependence and overall patterns in the data.

4 Work and Results

“One good test is worth a thousand expert opinions.”

– A sign at Boeing headquarters

“You can't make it better until you make it work.”

– Akin's 40th Law of Spacecraft Design (McBryan's Law)

In this chapter we will first revisit the findings of Subsection 2.1.3 and try to extract time-dependent failure behavior of satellites from today's in-flight reliability data. The major source of data for this chapter will be the work of Saleh & Castet [22], Dubos et al. [15, 118] and Castet & Saleh [115–117, 120]. Later, we will look into CubeSat reliability and analyze the results of our CFDB. In the last section of this chapter we will present the development of MOVE-II, its technical advancements, and the methods used to shift the risk of the project upfront. Methods to assess the reliability and analyze reliability growth on system level will be followed by an example how to use reliability prediction in future CubeSat missions as an additional tool for design trade-offs.

4.1 Analysis of Satellite Reliability

In general, determining the time-dependent failure behavior of past missions is an important task to reveal underlying patterns, and thus also causes, for satellite failure. This knowledge could also enhance prediction models for future missions. For example, the widely-used assumption of exponential distributions (constant failure rate) for satellite failures leads to different projections for masses of consumables necessary for missions than when using a Weibull failure rate [54]. Even within the Weibull distributions, the time-dependent failure rate will strongly depend on shape and scale parameter applied. Besides exponential distributions, Castet & Saleh [117] reported that Weibull distributions with a shape parameter of around 1.7 were commonly used in the past for satellite systems, representing an increasing failure rate for those missions. On the contrary, we have learned in Subsection 2.1.3 that most of the past space missions experienced infant mortality. Despite this, many mathematical reliability prediction models still use the constant failure rate approach.

Saleh & Castet already took a first step in solving this puzzle, and their analysis of a group of 1,584 satellites launched between January 1990 and October 2008 [22] is by far the most comprehensive study on this topic. On the next pages, we will focus on several results of their studies, including their mathematical models. Although some of their results are arguable, it should be noted that the overall scientific value of their work shall not be challenged by this discussion. A first analysis of their work was already presented in the Master's Thesis of Schummer [262], supervised by the author of this thesis. We will start with the mathematical models for overall satellite reliability, focusing on the models of the 2010 publication of Castet & Saleh [120], and their update on the models in a book in 2011 [22]. Later in the section, we will use the

models of Dubos et al. [15, 118], which differentiate between certain mass classes of spacecraft. In this section, Single-Weibull functions, 2-Weibull mixture functions, and modified Weibull extension distributions with a bathtub-shaped failure rate function [124] will be used to describe the on-orbit reliability of satellites over time. The reliability of the Single-Weibull function is defined as:

$$R(t) = \exp \left[- \left(\frac{t}{\theta} \right)^\beta \right] \quad (24)$$

where β is the aforementioned shape factor of the Weibull function, and θ the scale factor. The reliability within the 2-Weibull mixture function can be written as:

$$R(t) = \alpha_1 \cdot \exp \left[- \left(\frac{t}{\theta_1} \right)^{\beta_1} \right] + \alpha_2 \cdot \exp \left[- \left(\frac{t}{\theta_2} \right)^{\beta_2} \right] \quad (25)$$

where β_1 and β_2 are the shape factors of the Weibull functions, and θ_1 and θ_2 the scale factors. α_1 and α_2 are the mixing weights of the Weibull functions. Hence:

$$\alpha_1 + \alpha_2 = 1 \quad (26)$$

The mathematical description of the modified Weibull extension distribution with a bathtub-shaped failure rate function was already presented in Subsection 2.1.1. In [120], the overall reliability function of spacecraft is stated as a 2-Weibull mixture function with the following reliability function:

$$R(t) = 0.9484 \cdot \exp \left[- \left(\frac{t \text{ [y]}}{982,100} \right)^{0.2575} \right] + 0.0516 \cdot \exp \left[- \left(\frac{t \text{ [y]}}{10.2} \right)^{1.9970} \right] \quad (27)$$

Looking more closely at the parameters, the reliability of this studied group of satellites seems to be dominated by an infant mortality term ($\beta_1 = 0.2575$). The second term describes wear-out with an increasing concave failure rate ($1 < \beta_2 < 2$)⁶⁹ and influences the resulting function with a relatively mixing weight α_2 of slightly more than 0.05. The scale factor θ_1 of the infant mortality portion seems quite high with 982,100 years. In theory, θ describes the time when 63.21% of the studied group failed (compare Subsection 2.1.1). Clearly, that cannot be the case while using 982,100 years. Although such data should not be used to extrapolate lifetimes greater than the cut-off (which was 15 years in their case), other groups have done exactly that in the past with those results and came to reliability values of 80% after 100 years on-orbit [35], which is clearly not supported by the original data.

Despite this misuse, a closer look at the fractions of the reliability function within the studied timeframe reveals further issues. Figure 4-1 depicts the fraction of satellites failed due to the two terms of the 2-Weibull mixture function within the first seven years on-orbit. It can be noted that due to the scale factor θ_1 being very large, the infant mortality portion of the function continues to grow throughout the observation window. The wear-out portion, although starting quite early, looks realistic within the seven years' timeframe. Within the first year, 98.2% of all failed satellites failed due to the infant mortality term and only 1.8% due to the wear-out term, which also seems realistic. Expanding that timeframe to seven years, 69.3% of all failed satellites failed due to the infant mortality term and 30.7% due to the wear-out term⁷⁰.

⁶⁹ And being almost equivalent to the Raleigh distribution, thus linear increasing failure rate ($\beta = 2$).

⁷⁰ Overall, around 2.7% of all satellites failed within one year and around 6.3% within seven years on-orbit.

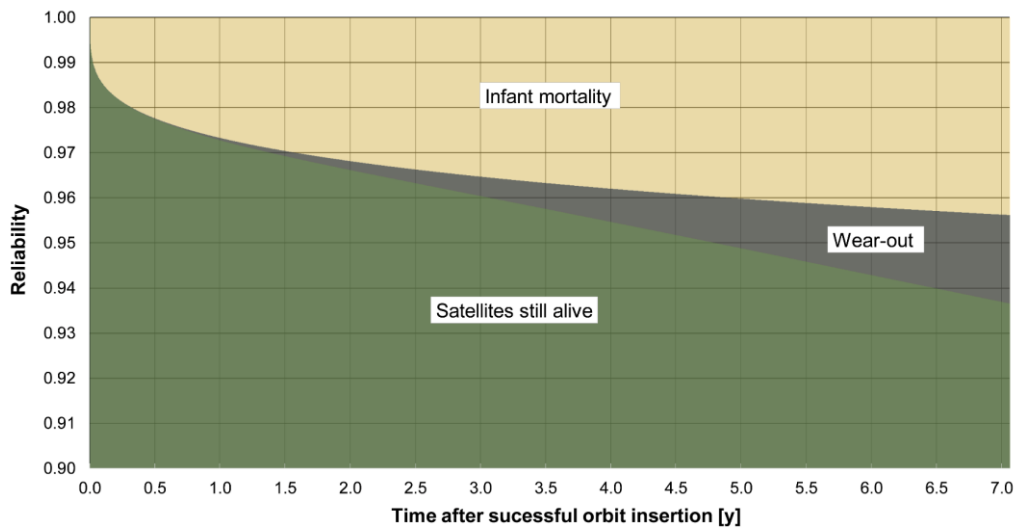


Figure 4-1: Fraction of satellites failed due to the two different portions of the 2-Weibull mixture function of Castet & Saleh [120] (equation (27)) within the first 7 years on-orbit. Green depicts satellite still alive, yellow satellites failed due to the infant mortality portion ($\beta_1 = 0.2575$) of the function, and black satellites failed due to the wear-out portion ($\beta_2 = 1.9970$) of the function.

Increasing that timeframe further to 14.5 years (the last failure point of the Castet and Saleh paper was at $t = 5,207$ days), it can be noted that the infant mortality fraction of the function increases over the whole lifetime (see Figure 4-2), although the failure rate itself is decreasing over time. This is again due to the extremely high value of θ_1 . The portion of satellites failing due to wear-out slowly decreases again after a turning point around $t = 7$ years. Figure 4-3 depicts the time behavior of the infant mortality term and Figure 4-4 the time behavior of the wear-out term, as a fraction of all satellites. Again, it can be noted that the infant mortality portion increases throughout the lifetime if modelled with the function by Castet & Saleh. Although there is no standardized value for the time period when infant mortality in space should flat out, other research suggests that values below one year of lifetime are realistic [98, 198, 200, 203, 207]. At 14.5 years, 54% of satellites that failed, failed due to the infant mortality portion of the function⁷¹. Figure 4-2 also reveals that a significant fraction of satellites failed due to the infant mortality term between year seven and year 14.5 which conflicts with the definition of infant mortality.

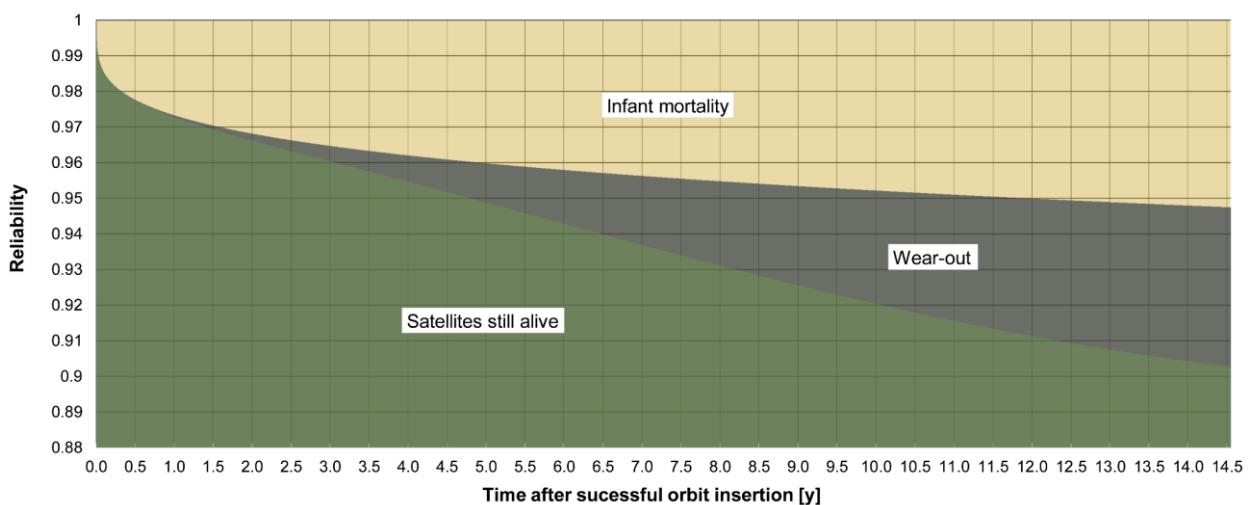


Figure 4-2: Fraction of satellites failed due to the two different portions of the 2-Weibull mixture function of Castet & Saleh [120] (equation (27)) within the first 14.5 years on-orbit. Green depicts satellite still alive, yellow satellites failed due to the infant mortality portion ($\beta_1 = 0.2575$) of the function, and black satellites failed due to the wear-out portion ($\beta_2 = 1.9970$) of the function.

⁷¹ Overall, around 9.7% of all satellites failed within 14.5 years on-orbit.

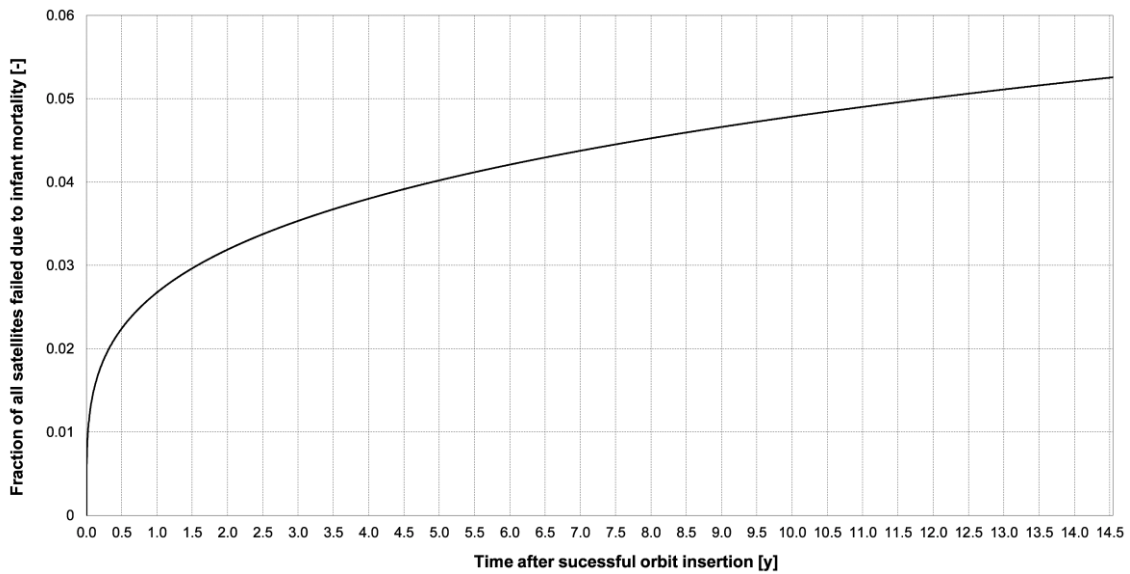


Figure 4-3: Fraction of all satellites failed due to the infant mortality term of the 2-Weibull mixture function of Castet & Saleh [120] (equation (27)).

Similarly, while looking at Figure 4-4, the wear-out effect modelled by the function of Castet & Saleh has some characteristics that need to be discussed. After the aforementioned turning point, the wear-out failure rate decreases until the end of the lifetime, an effect that can be attributed again to the scale factor of the term. With $\theta_2 = 10.2$ years, the majority of the wear-out group will have failed at that point in time. Combined with the increasing concave failure rate, this leads to overall less wear-out experienced by the group of satellites in year 14.5 than in year seven. In general, this could mean that a group of surviving satellites is more robust against certain wear-out effects in space than the satellites that failed earlier. However, effects such as TID and mechanical wear-out of reaction wheels oppose that. Also it should be noted that out of the group of 1,584 satellites, 1,486 were censored [116], meaning that they either were retired before failure occurred or were still operational at the end of the observation window. Thus, at the end of the studied timeframe, data have to be handled with care due to the potentially reduced number of still working satellites with respect to failed ones.

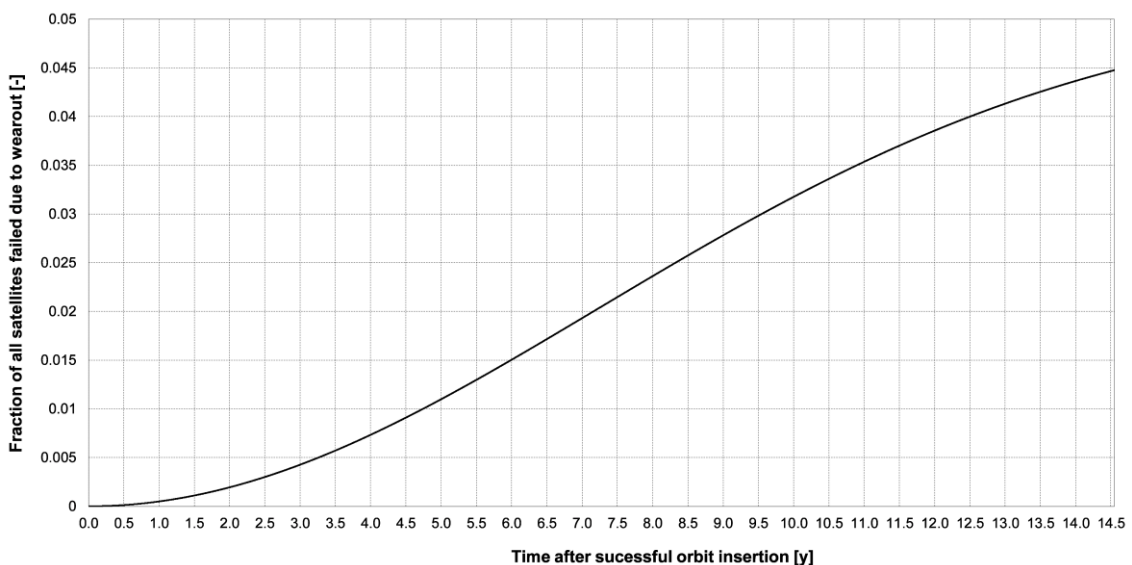


Figure 4-4: Fraction of all satellites failed due to the wear-out term of the 2-Weibull mixture function of Castet & Saleh [120] (equation (27)).

To summarize, although the 2-Weibull mixture function is within 0.55% of the maximum error to the nonparametric estimation [120], the study of the terms of the function reveals that the fit uses on-orbit behavior of satellites that cannot be fully explained by the author of this thesis (i.e., infant mortality over the whole studied timeframe, wear-out fraction that decreases towards the end).

In an update in 2011 [22], both authors presented a new 2-Weibull mixture function with the following parameters:

$$R(t) = 0.9725 \cdot \exp \left[- \left(\frac{t \text{ [y]}}{14,310.1} \right)^{0.3760} \right] + 0.0275 \cdot \exp \left[- \left(\frac{t \text{ [y]}}{9.3} \right)^{2.9937} \right] \quad (28)$$

In this 2-Weibull mixture function, the dominant portion is again infant mortality, described by an increasing failure rate of $\beta_1 = 0.3760$ and influencing the overall function by a factor of more than 97%. Again, the scale factor of the infant mortality term is quite large ($\theta_1 = 14,310$ years) although it is almost two magnitudes lower than the one in their 2010 fit. In the updated function, the wear-out term has a larger shape factor ($\beta_2 = 2.9937$), describing an increasing convex failure rate. The scale factor of the wear-out term is in the same order of magnitude as before ($\theta_2 = 9.3$ years). As it can be seen in Figure 4-5, the infant mortality portion after seven years is even larger than in the 2010 fit. After one year on-orbit, almost all (99.9%) of the satellites that have failed, failed due to the infant mortality portion of the function. At $t = 7$ years, still 85% of all failed satellites failed due to the infant mortality term of the function⁷². Both can be attributed mainly to a lower scale factor of the infant mortality portion, and a slightly larger proportion of the first term in the overall fit. The wear-out portion remains relatively small within the seven years' timeframe, as we would expect from a function with higher shape factor and an only slightly lower scale factor than the original one.

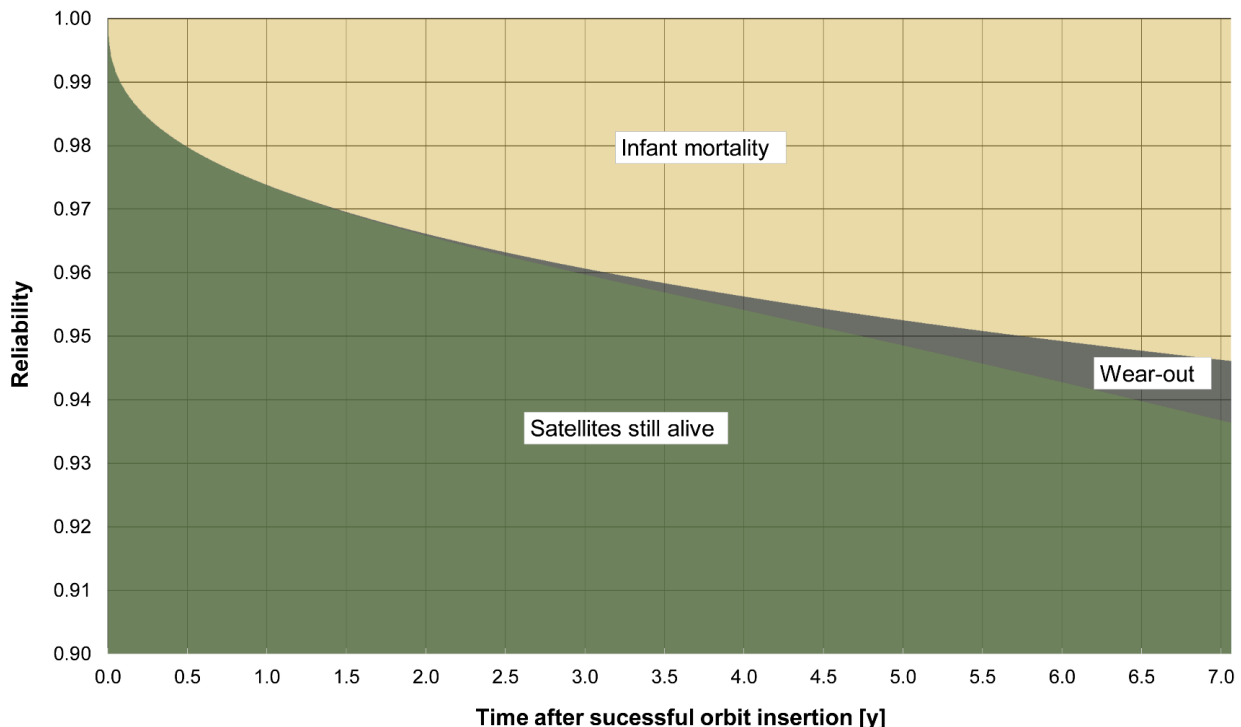


Figure 4-5: Fraction of satellites failed due to the two different portions of the 2-Weibull mixture function of Saleh & Castet [22] (equation (28)) within the first 7 years on-orbit. Green depicts satellites still alive, yellow satellites failed due to the infant mortality portion ($\beta_1 = 0.3760$) of the function, and black satellites failed due to the wear-out portion ($\beta_2 = 2.9937$) of the function.

⁷² Similar to before, around 2.6% of all satellites failed within one year and around 6.3% within seven years on-orbit.

Extending the observation window to $t = 14.5$ years (see Figure 4-6), a less pronounced wear-out, and thus a more pronounced infant mortality with respect to Figure 4-4 can be observed. Of the 9.7% of satellites that have failed until 14.5 years, 72% failed due to the infant mortality term of the function. Similarly, to the old function, many satellites fail due to the infant mortality term between year seven and the end of the observation window, contradicting the meaning of infant mortality.

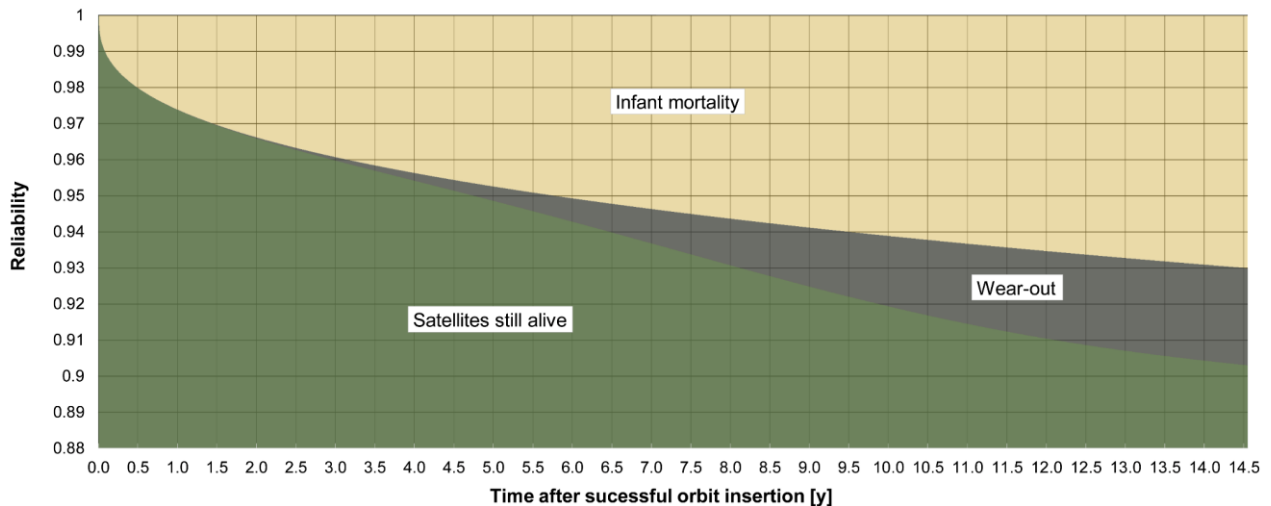


Figure 4-6: Fraction of satellites failed due to the two different portions of the 2-Weibull mixture function of Saleh & Castet [22] (equation (28)) within the first 14.5 years on-orbit. Green depicts satellite still alive, yellow satellites failed due to the infant mortality portion ($\beta_1 = 0.2575$) of the function, and black satellites failed due to the wear-out portion ($\beta_2 = 1.9970$) of the function.

Looking again more closely at the infant mortality portion (Figure 4-7) and the wear-out portion (Figure 4-8), similarities to the 2010 fit (equation (27)) can be found. The infant mortality decreases over time, but satellites are still failing due to the infant mortality term late in the lifetime. Again, this can be attributed to the scale factor of the infant mortality term, which describes that 36.79% of the satellites have not failed at $t = 14,310$ years. Of course, an extrapolation up to this point is not valid, but inconsistencies due to the large scale factor within the observation window must be addressed. Also, the infant mortality term decreases less than in the 2010 fit. That can at least be partly attributed to the increased beta factor, which bends the curve more towards the constant failure rate (while not nearly approaching it with $\beta_1 = 0.3760$).

The wear-out fraction, depicted in Figure 4-8, is more S-shaped than in the earlier fit, meaning that the wear-out modelled by the function decreases after $t = 8$ years. As noted before, although it could be caused by an unknown phenomenon, wear-out of satellites should not decrease over the satellite's lifetime in the opinion of the author. This S-shape is caused by the slightly smaller scale factor, having 63.21% of all wear-out failures already at $t = 9.3$ years, and the slightly increased shape factor, yielding in overall steeper increase and decrease of the function.

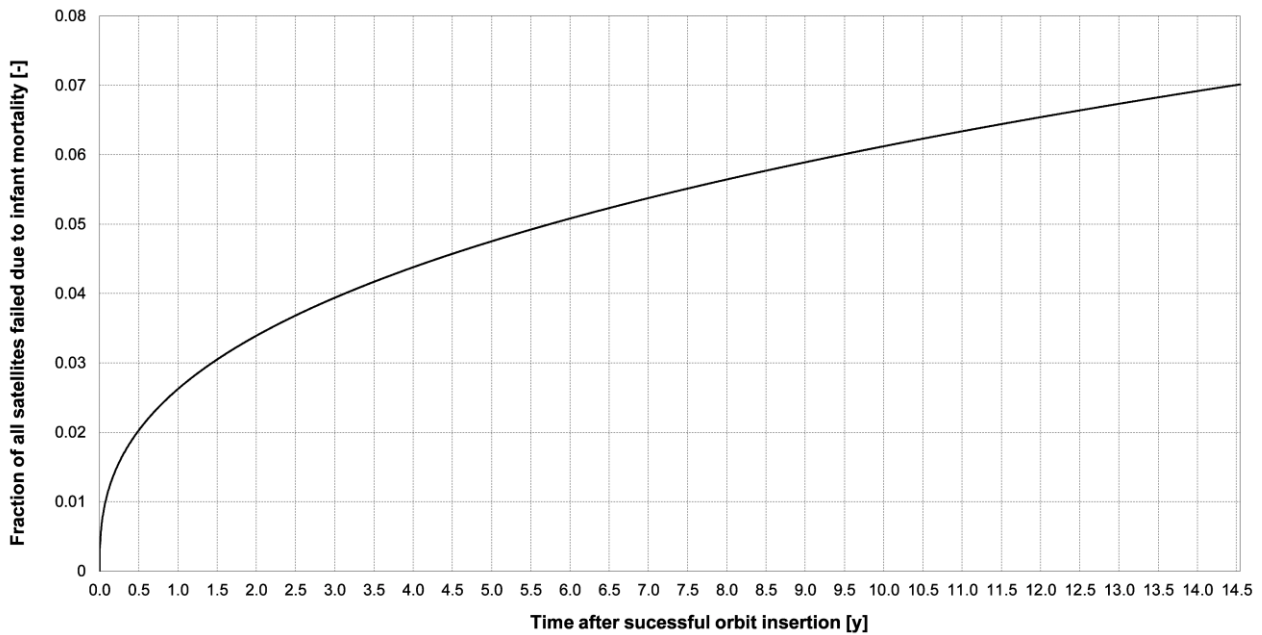


Figure 4-7: Fraction of all satellites failed due to the infant mortality term of the 2-Weibull mixture function of Saleh & Castet [22] (equation (28)).

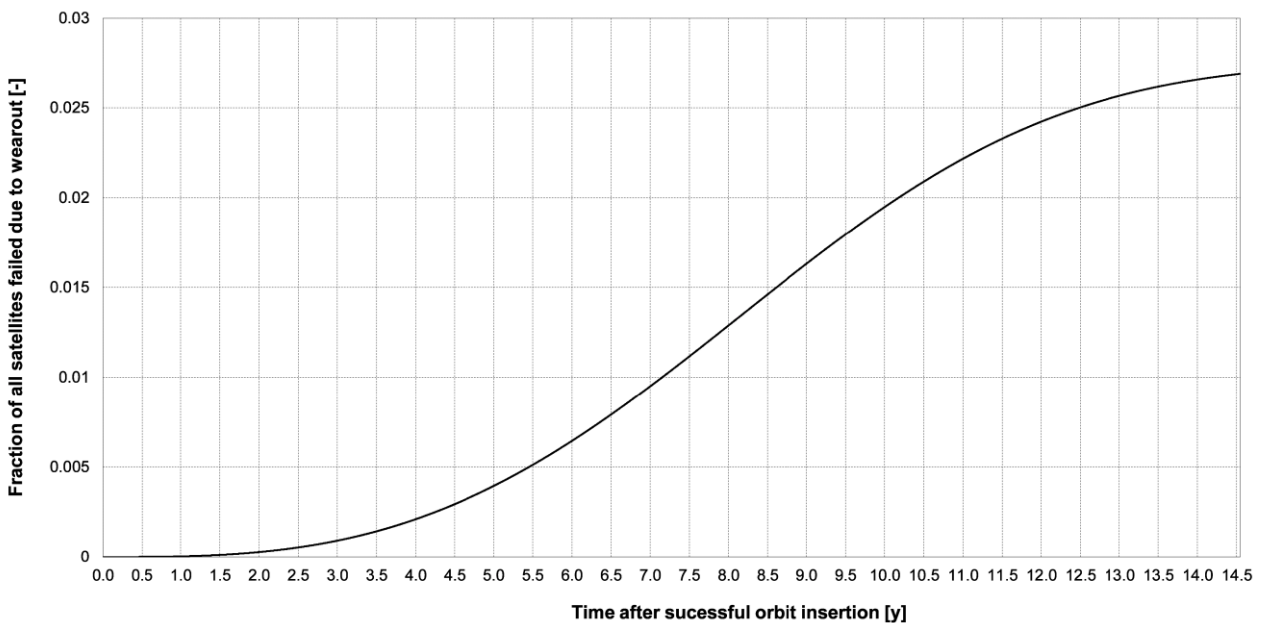


Figure 4-8: Fraction of all satellites failed due to the wear-out term of the 2-Weibull mixture function of Saleh & Castet [22] (equation (28)).

To summarize, the mathematical descriptions show inconsistencies not explainable by the author of this thesis. While having no access to the underlying reasons behind each satellite failure, it seems to be more valid to generally treat satellites as any other technical system – with early infant mortality, followed by a constant failure rate and middle-to-late wear-out phenomena, and not the other way around. Also, looking at the relative contributions of each subsystem in this studied group of satellites [117], one can clearly distinguish between subsystems affected by DOA and infant mortality (Thruster, Solar-Array Deployment (SAD), Mechanisms), subsystems middle-to late wear-out (Battery) and subsystems with near-constant failure rate or with no clear pattern.

Thus, to study the overall time-dependent failure behavior of this group of satellites, a modified Weibull extension distribution with a bathtub-shaped failure rate function (see equation (12)), similar to the one presented in the work of Peng & Zhang [124], was used as a first approach in order to shed light on the overall shape of the failure rate function over time. Thereby, the reliability function was estimated by a nonlinear least squares approach as:

$$R(t) = \exp \left\{ 0.07437 \cdot 0.08466 \cdot \left\{ 1 - \exp \left[\left(\frac{t \text{ [y]}}{0.08466} \right)^{0.2033} \right] \right\} \right\} \quad (29)$$

To evaluate the resulting fit, the Kaplan-Meier nonparametric estimation of Castet & Saleh [115] was used up to a cut-off time of $t = 14.5$ years. This cut-off was chosen since the original nonparametric data are only considered valid up to the point of the last failure date, thus the Kaplan-Meier estimated nonparametric curve ends at $t = 14.256$ years (last failure date) as done before and suggested by Saleh & Castet [22] based on publications by Kalbfleisch & Prentice. These nonparametric values, including the 95% confidence intervals, can be found in Appendix B, Table 6-7. The resulting fit stays within a maximum error of 0.64 percentage points to the nonparametric estimation (see Figure 4-9).

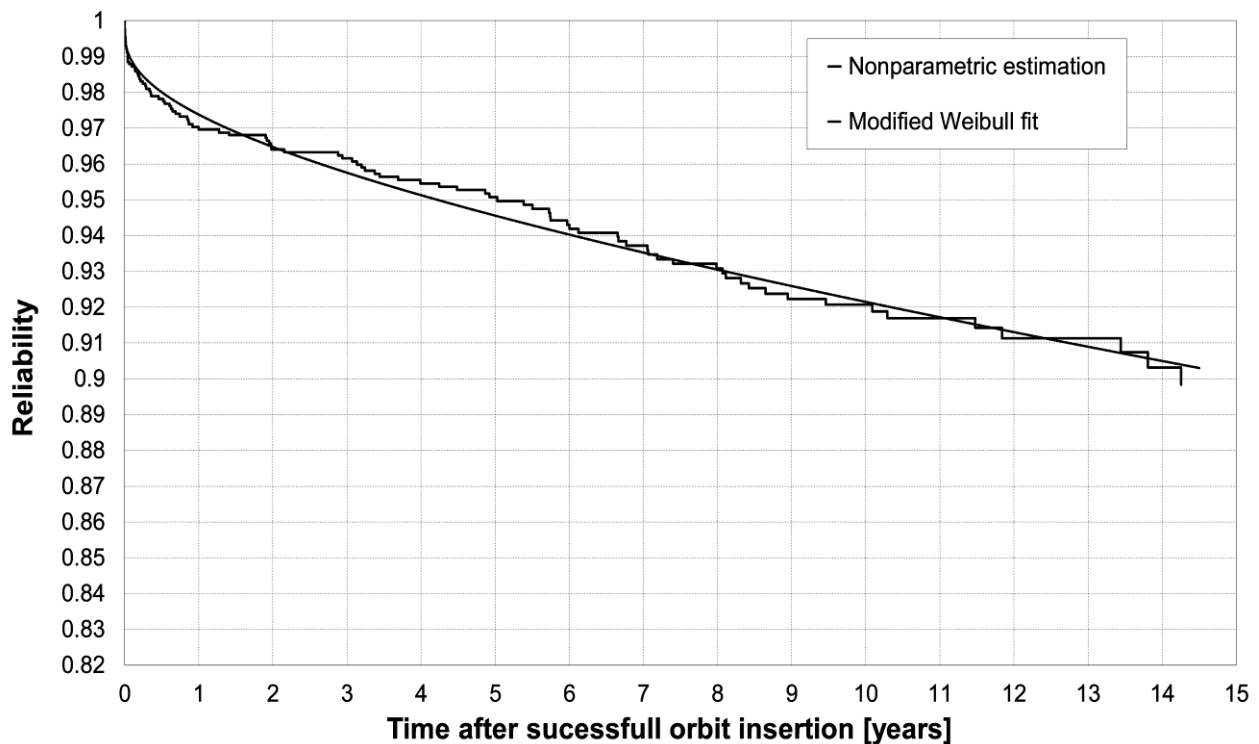


Figure 4-9: Modified Weibull extension fit with a bathtub-shaped failure rate function (equation (29)) and the underlying nonparametric estimation. Data source of nonparametric estimation: [115].

With a goodness-of-fit value of $R^2 = 0.988$, the modified Weibull fit follows the nonparametric estimation very well as it can also be seen in the Weibull-plot (Figure 4-10, left) and the dispersion of the boxplot (Figure 4-10, right). As depicted in the Weibull-plot, the parametric estimation generally aligns with the nonparametric one, except for very early failures (the first three points correspond to failures on the first, second and third day). The 25th percentile (-0.268%) and the 75th percentile (0.167%) of the residuals show that the modified Weibull fit is gradually more dispersed than the first Castet & Saleh 2-Weibull mixture function (-0.13% and 0.15%) [120] as well as the later Saleh & Castet 2-Weibull mixture function (-0.14% and 0.16%) [22].

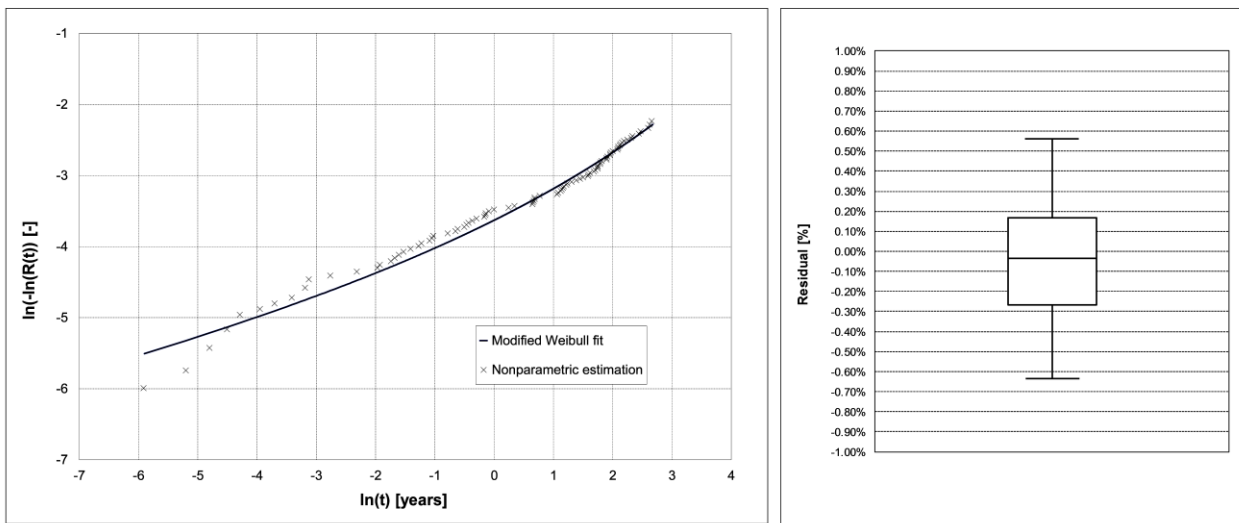


Figure 4-10: Weibull-plot (left) and boxplot of the residuals between the modified Weibull fit (equation (29)) and the nonparametric estimation. Data source of nonparametric estimation: [115]

The origin of this dispersion can be seen when plotting all three fits, as depicted in Figure 4-11. The modified Weibull fit mainly deviates in two to three regions from the 2-Weibull mixture models, although the deviation is always less than 0.5 percentage points, as can be seen in Figure 4-12.

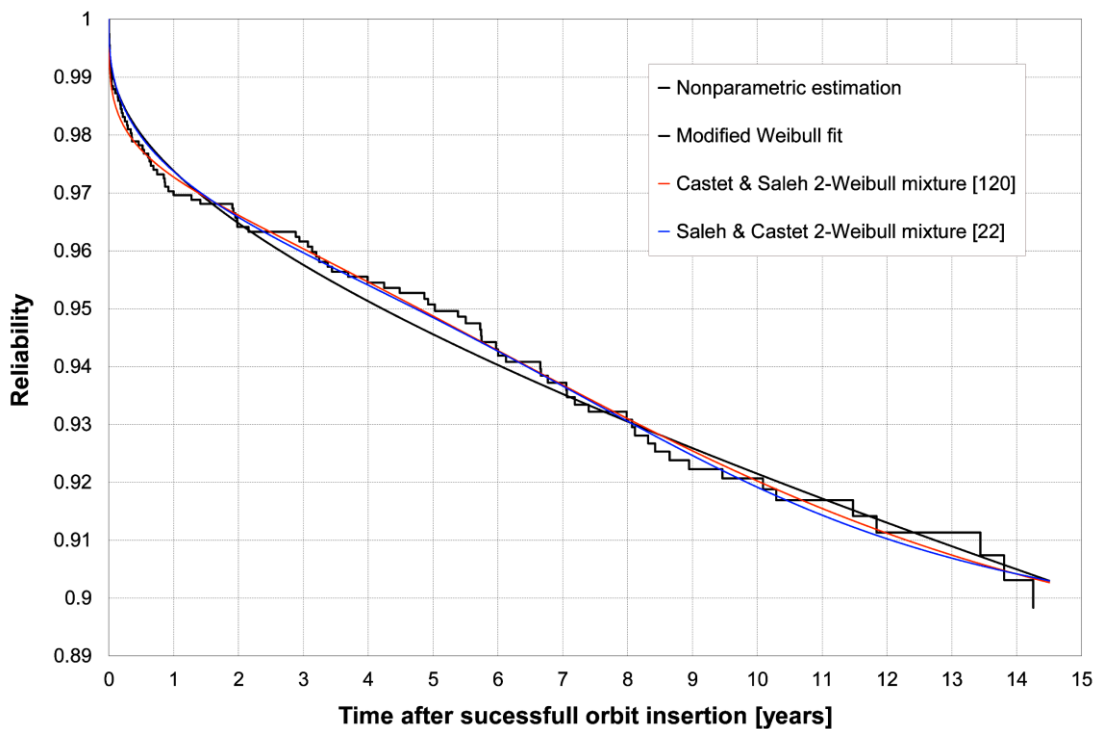


Figure 4-11: Modified Weibull (equation (29)) fit vs. 2-Weibull mixture models of Castet & Saleh [120] (equation (27)) and Saleh & Castet [22] (equation (28)). Data source of nonparametric estimation: [115].

Up to $t = 1.5$ years, the first fit by Castet & Saleh [120] follows the nonparametric estimation best, and both the modified Weibull fit as well as the newer Saleh & Castet fit [22] slightly overestimate satellite reliability during that time. Later, two regions with more pronounced deviations of the modified Weibull fit from the nonparametric estimation and the two other parametric fits can be seen. The values over- and undershoot the estimation by the modified Weibull function in these regions. This suggests that the group of satellites must be studied in more detail for other underlying causes.

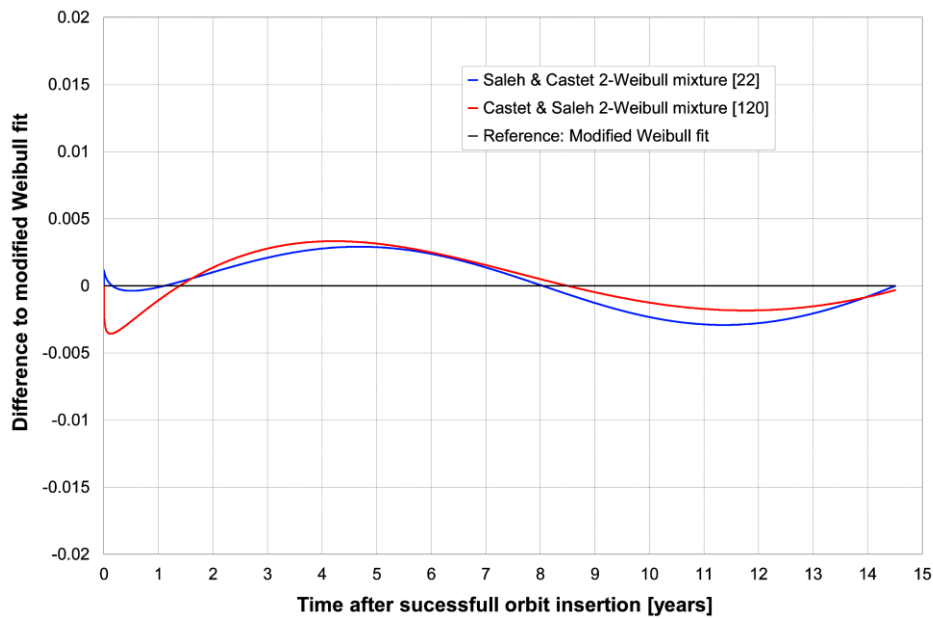


Figure 4-12: Difference of the 2-Weibull mixture models of Castet & Saleh [120] (equation (27)) and Saleh & Castet [22] (equation (28)) to the modified Weibull fit (equation (29)).

Most interesting, the failure rate of the modified Weibull function shows no clear pattern of wear-out in the underlying data, as can be seen in Figure 4-13. The bathtub “does not hold water” within the observed timeframe, thus wear-out is either too scarce to be detected or masked. If $\beta < 1$, a change point, i.e., the point in time when the failure rate starts growing again, can be estimated with [124]:

$$t^* = \alpha \cdot \left(\frac{1}{\beta} - 1\right)^{\frac{1}{\beta}} \quad (30)$$

For this fit, the change point is at approximately $t^* = 70$ years. This emphasizes the missing wear-out within the observation window, and as before, values beyond the observation window shall not be used for any extrapolation.

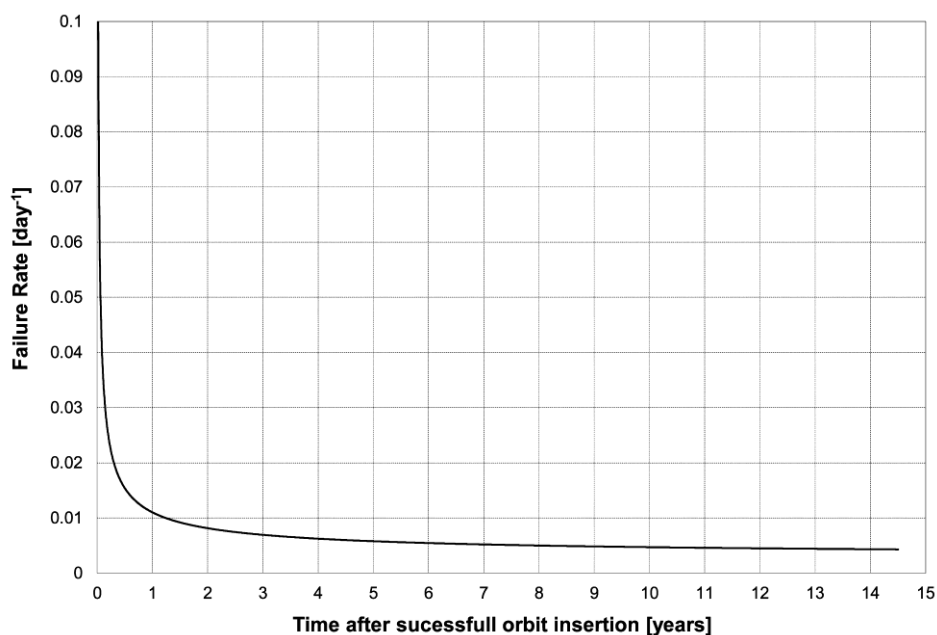


Figure 4-13: Failure rate of the modified Weibull function.

Again, without having access to the underlying causes for failure, we can only speculate about the reason for that. It might be the case that many satellites were retired before they failed due to wear-out, while many of the infant mortality failures cases were not prevented by this, as they occur rather surprisingly for the satellite's owner in most cases. As can be seen in Figure 4-14, the modified Weibull function found in this work only slightly deviates from the one found by Peng & Zhang [124]. Their failure rate function also shows a right-hand open bathtub shape (see Figure 6-1 in Appendix B).

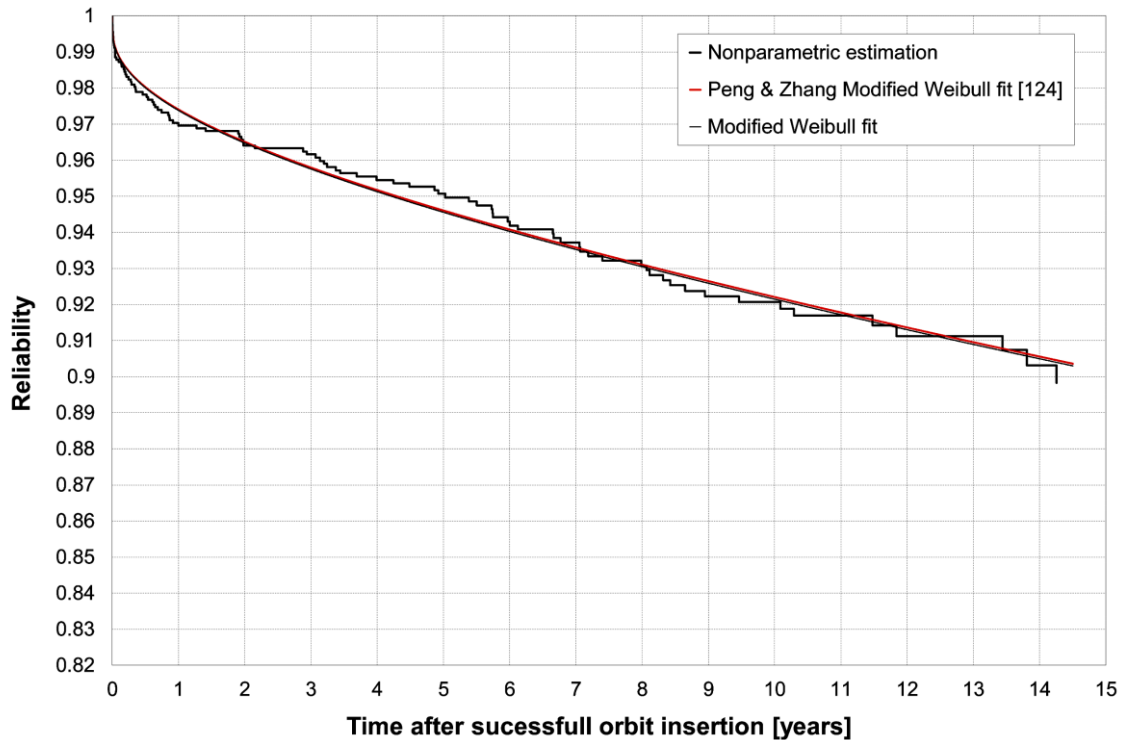


Figure 4-14: Modified Weibull fit (equation (29)) vs. modified Weibull fit by Peng & Zhang [124]. Data source of nonparametric estimation: [115].

To further study the different contributions to on-orbit failure of satellites, a set of 2-Weibull mixture functions was fitted to the nonparametric data, starting with a 2-Weibull mixture model in which the starting point of the shape factor β_2 for the nonlinear least squares estimation was set to 1. This results directly from the above mentioned right-open bathtub shape of the modified Weibull fit, i.e., the infant mortality being pronounced in the mixed group of small-, medium- and large-sized satellites than wear-out effects. The 95% confidence interval of the shape factor β_2 for this 2-Weibull mixture function was almost equally dispersed around one (0.8769, 1.123). Thus, a new 2-Weibull mixture fit where the β_2 parameter was held constant to a value of one was done subsequently. The resulting reliability function was estimated as:

$$R(t) = 0.02377 \cdot \exp \left[- \left(\frac{t [y]}{0.1589} \right)^{0.5274} \right] + 0.97623 \cdot \exp \left[- \left(\frac{t [y]}{169.6} \right)^1 \right] \quad (31)$$

Although the infant mortality portion of the overall fit seems to be relatively small at first glance ($\alpha_1 = 0.02377$, with a 95% confidence interval of 0.01865 and 0.02889), almost 23% of all satellites that failed within the observation window of 14.5 years failed due to infant mortality, as can be seen in Figure 4-16. Different from the 2-Weibull mixture models of Castet & Saleh, this fit comprises scale factors that are more appropriate to the specific contribution they should represent. The infant mortality portion of the function is built by a shape factor of $\beta_1 = 0.5274$ (95% confidence interval: 0.4189, 0.6359) and a scale factor of $\theta_1 = 0.1589$ years (95% confidence interval: 0.0503 years, 0.2674 years). Thus, the infant mortality portion of the

function will fade out quickly after the first year on-orbit, as depicted in Figure 4-17. The goodness-of-fit value of the overall function is $R^2 = 0.9955$ and the 2-Weibull mixture model follows the nonparametric estimation better than the modified Weibull function, as can be seen in Figure 4-15.

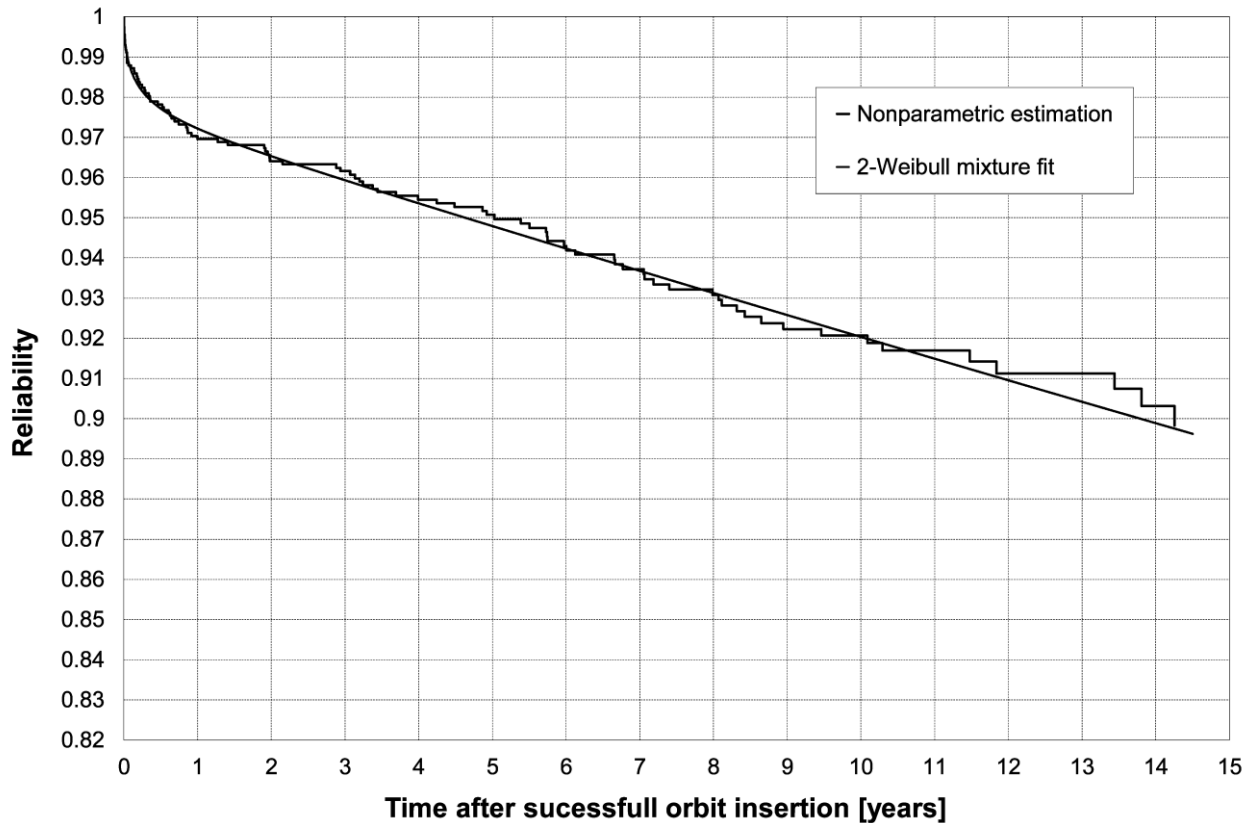


Figure 4-15: 2-Weibull mixture fit (equation (31)) and underlying nonparametric estimation. Data source of nonparametric estimation: [115].

The second part of the fit is built by a function with constant failure rate, thus describing failure within the regular life of the satellites. Although the shape factor of the second function is $\beta_2 = 1$, more than one underlying cause might be mixed in this portion of the overall fit. Since different classes of satellites are binned together in the studied group, it could be that the wear-out phase of one class of satellites (for example small satellites) mixes with non-failure of another class (for example large satellites) to a near constant failure rate over time. Also, it can be noted that, similar to the modified Weibull function, little to no wear-out can be seen until the end of the observation window, although such effects would be expected from subsystems such as batteries, as mentioned before.

Without the underlying detailed failure data, we can only speculate that the wear-out effect of certain subsystems is masked by the binning of different mass-classes of satellites, the aforementioned retirement of satellites and the general on-orbit failure behavior of satellites (i.e., satellites that survive the first 100 or 150 days on-orbit can often exceed their planned life, as described in Subsection 2.1.3). The second part of the 2-Weibull mixture function is completed with a scale factor of $\theta_2 = 169.6$ years (95% confidence interval: 124.9 years, 214.2 years) and a mixture weight α_2 of 0.97623 (95% confidence interval: 0.97111 and 0.98135). As depicted in Figure 4-16, this means that the fraction of satellites failed due to this part of the function steadily increases over time. As the parametric model is only valid up to the point of the last satellite failure in the database, satellite reliability after that point cannot be predicted. Therefore, although $\theta_2 = 169.6$ years would mean that the model predicts a significant number of satellites still alive at $t = 169.9$ years, $t = 14.256$ years is the latest point in time valid for predictions.

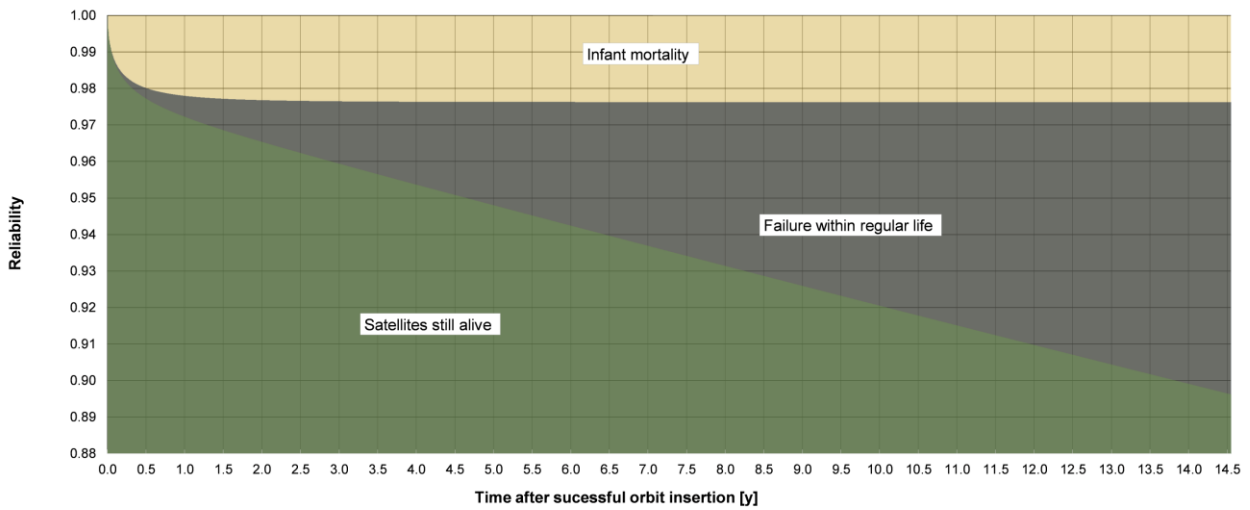


Figure 4-16: Fraction of satellites failed due to the two different portions of the 2-Weibull mixture function (equation (31)) within 14.5 years. Green depicts satellites still alive, yellow satellites failed due to the infant mortality portion ($\beta_1 = 0.5274$) of the function, and black satellites failed due to the constant failure rate portion ($\beta_2 = 1$) of the function.

Overall, around 2.4% of all satellites failed due to the first Weibull function of the model, representing early failures, and around 8% due to the second Weibull function. As stated before, the second Weibull term could be a mixture of different contributions, which could be broadly summarized as failure within regular life. Again, the 25th percentile (-0.27%) and the 75th percentile (0.07%) of the residuals are a little bit more dispersed than the Castet & Saleh 2-Weibull mixture function (-0.13% and 0.15%) [120] as well as the Saleh & Castet 2-Weibull mixture function (-0.14% and 0.16%) [22], as can be seen in the Weibull-plot (Figure 4-18 left) and the dispersion of the boxplot (Figure 4-18 right).

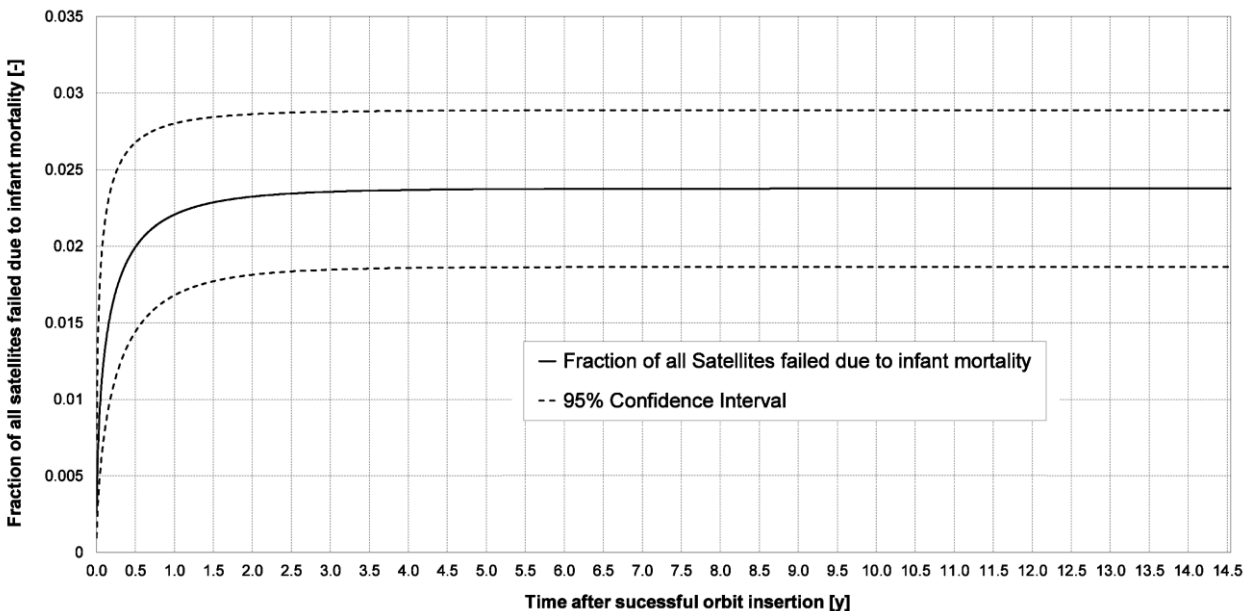


Figure 4-17: Fraction of all satellites failed due to the infant mortality term of the 2-Weibull mixture function (equation (31)).

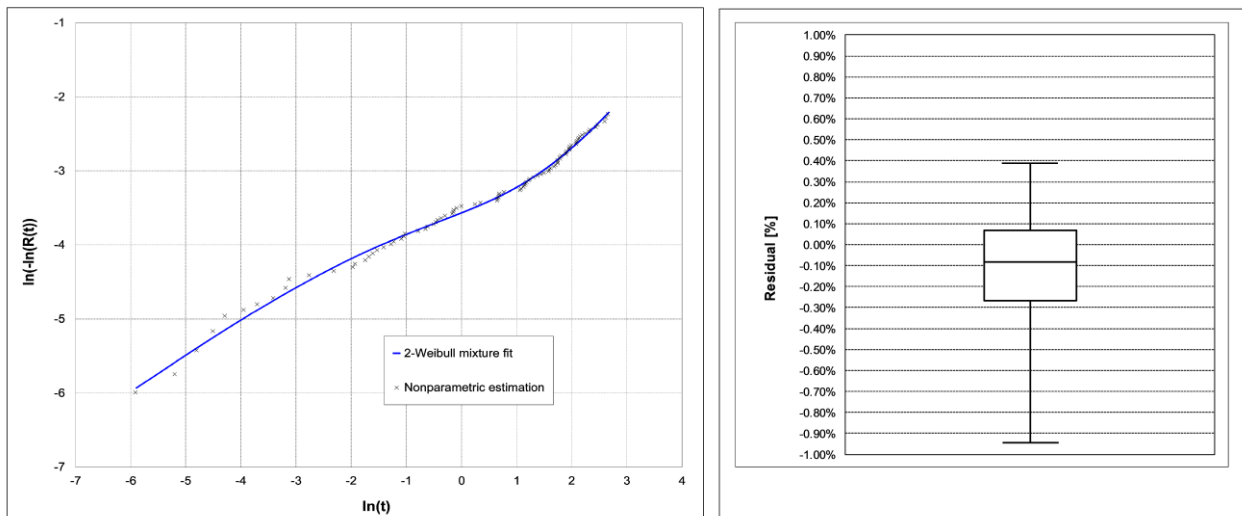


Figure 4-18: Weibull-plot (left) and boxplot of the residuals between the 2-Weibull mixture fit (equation (31)) and the nonparametric estimation. Data source of nonparametric estimation: [115]

As can be seen in Figure 4-19 and Figure 4-20, the deviations to the Kaplan-Meier estimation largely concentrate in the late part of the observation window, in which the parametric fit underestimates the on-orbit reliability. Overall, the deviations to the Weibull functions of Castet & Saleh are less than with the modified Weibull function, and again mainly located in later stages of the observation window. Figure 4-19 shows a comparison of the three models to the nonparametric estimation. Figure 4-20 depicts the deviation of the two functions by Castet & Saleh to the new 2-Weibull mixture model. The reduced number of failures, i.e., the positive deviation between 11 and 14 years to the nonparametric estimation and the two other models cannot be fully explained by physical or technical reasons. It could be the case that again the binning of different classes of satellites combined with the high rate of censored (i.e., retired, or limited due to the end of the observation window) satellites leads to this deviation. Also, the cut-off of the nonparametric data at the latest point of failure (at $t = 14.256$ years) might influence the parametric fit at late observation points.

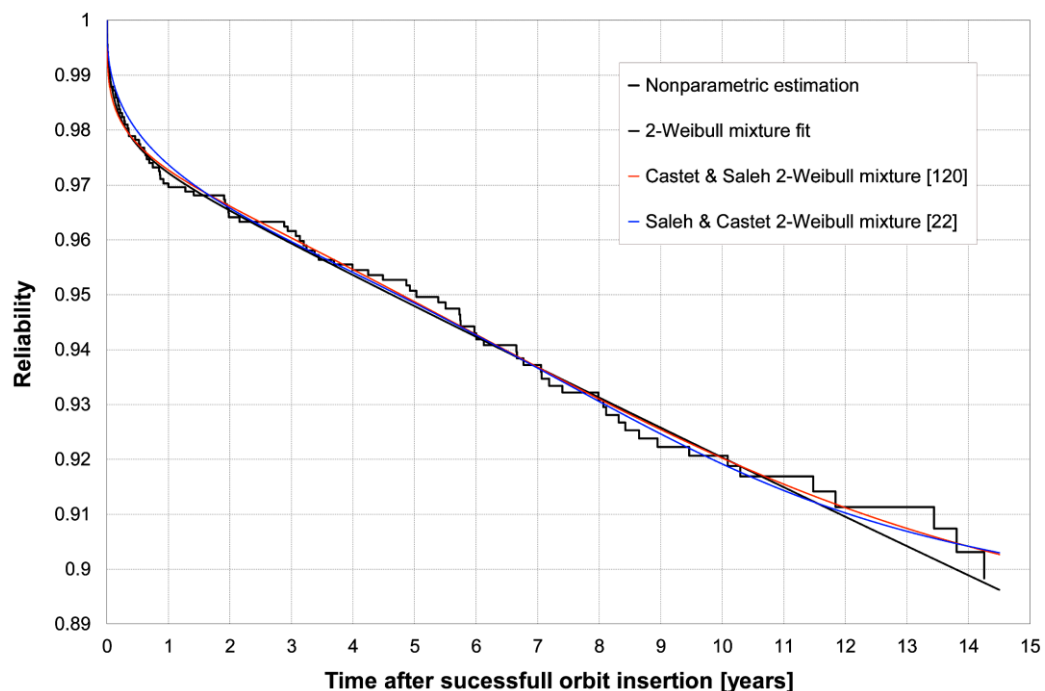


Figure 4-19: 2-Weibull mixture fit (equation (31)) vs. 2-Weibull mixture models of Castet & Saleh [120] (equation (27)) and Saleh & Castet [22] (equation (28)). Data source of nonparametric estimation: [115].

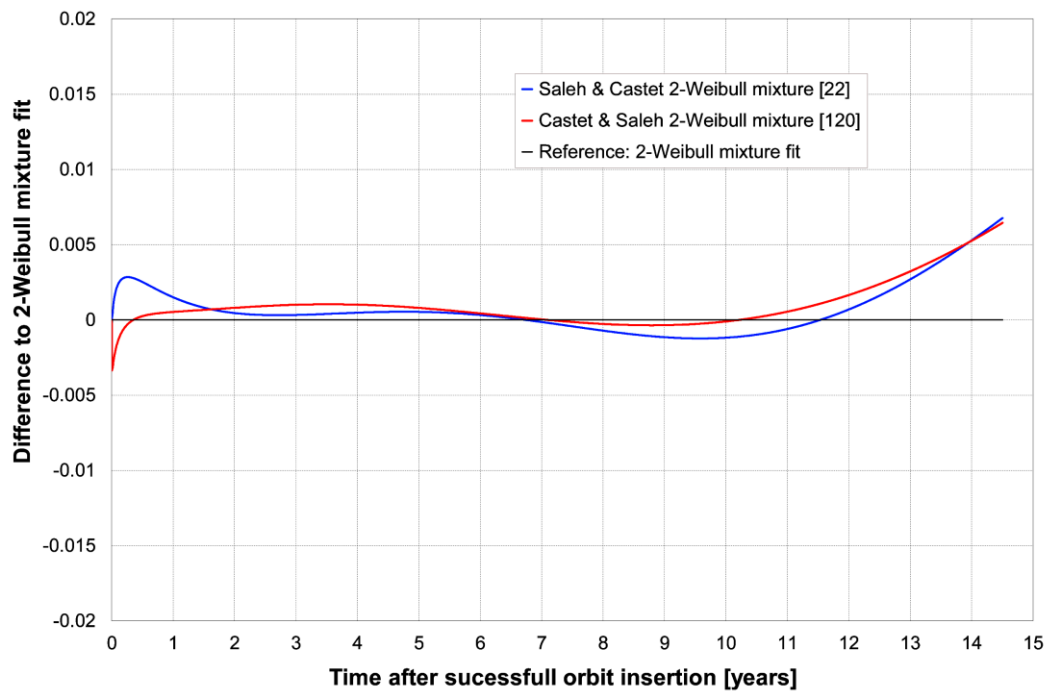


Figure 4-20: Difference of the 2-Weibull mixture models of Castet & Saleh [120] (equation (27)) and Saleh & Castet [22] (equation (28)) to the 2-Weibull mixture fit (equation (31)).

Looking at the underlying nonparametric data (Appendix B, Table 6-7), it can be observed that multiple satellites failed within their first week on-orbit. The question arises if those satellites ever were in a functional state after orbit insertion or were DOA. For terrestrial applications this is also called zero-times failures or out-of-the-box failures and describes produced items arriving in a non-functional state. Usually, many parametric fits start with a reliability value of 100% at $t = 0$. To cope with the DOA cases, the 2-Weibull mixture function (and other parametric functions) must be modified, as the failure time of the out-of-the-box failures is zero. To do so, the so-called Percent-Non-Zero (PNZ) calculation factor was introduced, as it can handle those failed items [263]. The before shown 2-Weibull mixture function is multiplied by the ratio of non-zero failure items (called p_{NZ}):

$$R(t) = p_{NZ} \cdot \left(\alpha_1 \cdot \exp \left[- \left(\frac{t}{\theta_1} \right)^{\beta_1} \right] + \alpha_2 \cdot \exp \left[- \left(\frac{t}{\theta_2} \right)^{\beta_2} \right] \right) \quad (32)$$

Thus, the satellites that never were in a functional state can also be handled by the equation. Without the detailed information about the root cause of each satellite failure, it can only be speculated which satellites were never in a functional state after on-orbit insertion, and which simply failed due to other reasons within the first week. Since the first week often involves all the necessary checkouts and hardware/software activation, the satellites might fail not directly on arrival, but experience their death only after certain functions are activated. On the other hand, some satellites might experience an early failure shortly after orbit insertion although they were fully functional. It might be a mixture of both factors, and the detailed root causes could shed light on that problem. As a next step in this work, a p_{NZ} value was introduced in the 2-Weibull mixture function (with $\beta_2 = 1$), and left flexible to guide us in some direction regarding the satellites that were dead on arrival/activation. The nonlinear least-squares estimated fit was:

$$R(t) = 0.9941 \cdot \left(0.0171 \cdot \exp \left[- \left(\frac{t [y]}{0.2641} \right)^{0.9342} \right] + 0.9829 \cdot \exp \left[- \left(\frac{t [y]}{164.8} \right)^1 \right] \right) \quad (33)$$

The resulting function (see Figure 4-21) shows a good alignment with the nonparametric data and has goodness-of-fit value of $R^2 = 0.995$. The p_{NZ} value of 0.9941 (95% confidence interval: 0.9922, 0.9961) means, that 0.59% (0.78%, 0.39%) of all satellites are estimated to never have been in a functional state after orbit-insertion. Work on the relative contributions of each subsystem in the studied group by Castet & Saleh [117] gives some examples for which this dead-on-activation behavior might be applicable (Solar-Array Deployment, Mechanisms). It can be observed that the value of β_1 increased with respect to the other 2-Weibull mixture functions, and is now near the constant failure rate value. The p_{NZ} factor led to insignificant changes of all other values of the parametric fit but resulted in narrower 95% confidence intervals⁷³. This can be, at least partly, attributed to the additional fitting parameter in the equation.

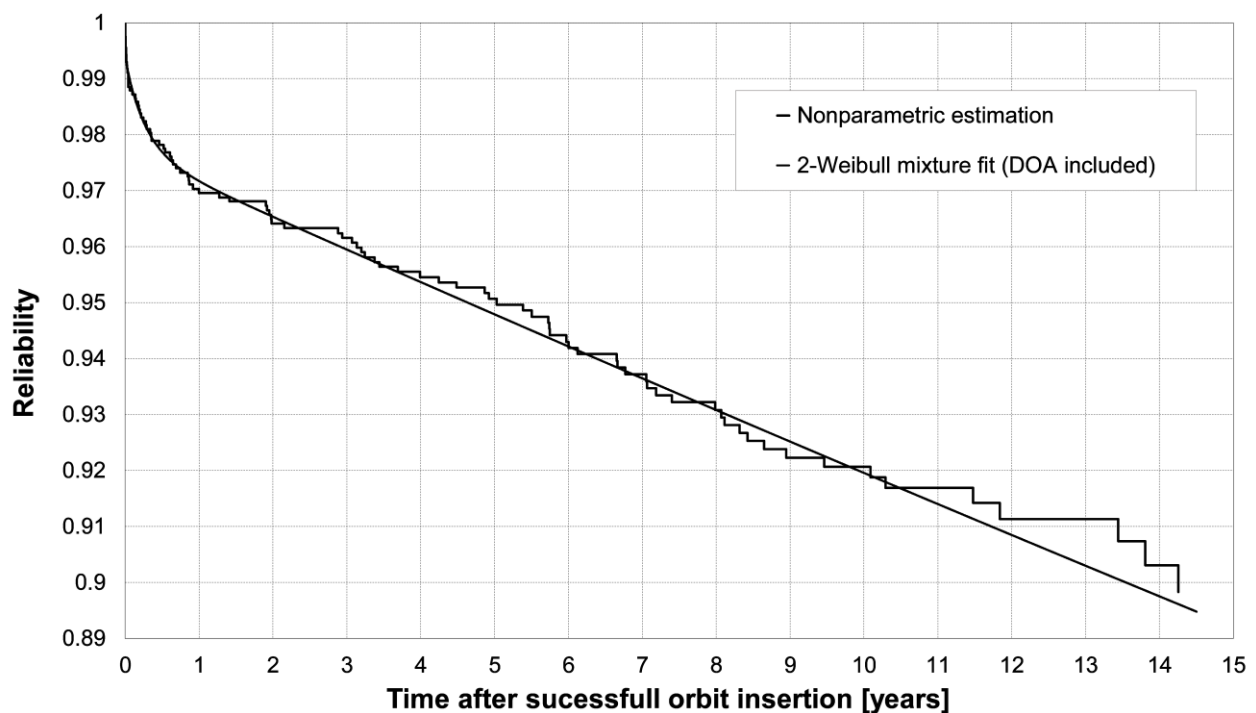


Figure 4-21: PNZ-modified 2-Weibull mixture fit (equation (33)). Data source of nonparametric estimation: [115].

Again, the largest fraction of failed satellites described by this parametric fit is approximated with a constant failure rate and thus fails within their lifetime (see Figure 4-22). After one year, approximately 1.6% of all satellites failed due to the infant mortality term of the function, in addition to the 0.59% that failed directly after orbit insertion. At the end of the observation window, around 2.3% of all satellites failed due to these two terms. This deviates less than 0.1% from the prediction of the other two presented 2-Weibull mixture functions. After 14.5 years, 8.2% of all satellites will have failed due to a failure described by the second term, i.e., constant failure rate. The p_{nz} factor helps to introduce satellites that were never functional into the parametric fit, described as “Death on Arrival” in Figure 4-22. Dead on arrival can also be understood as “dead on activation”, as already pointed out.

⁷³ 95% confidence intervals: $\alpha_1 = (0.01471, 0.01949)$, $\alpha_2 = (0.98051, 0.98529)$, $\beta_1 = (0.6191, 1.249)$, $\theta_1 = (0.1996 \text{ years}, 0.3285 \text{ years})$, $\theta_2 = (160.4 \text{ years}, 169.2 \text{ years})$

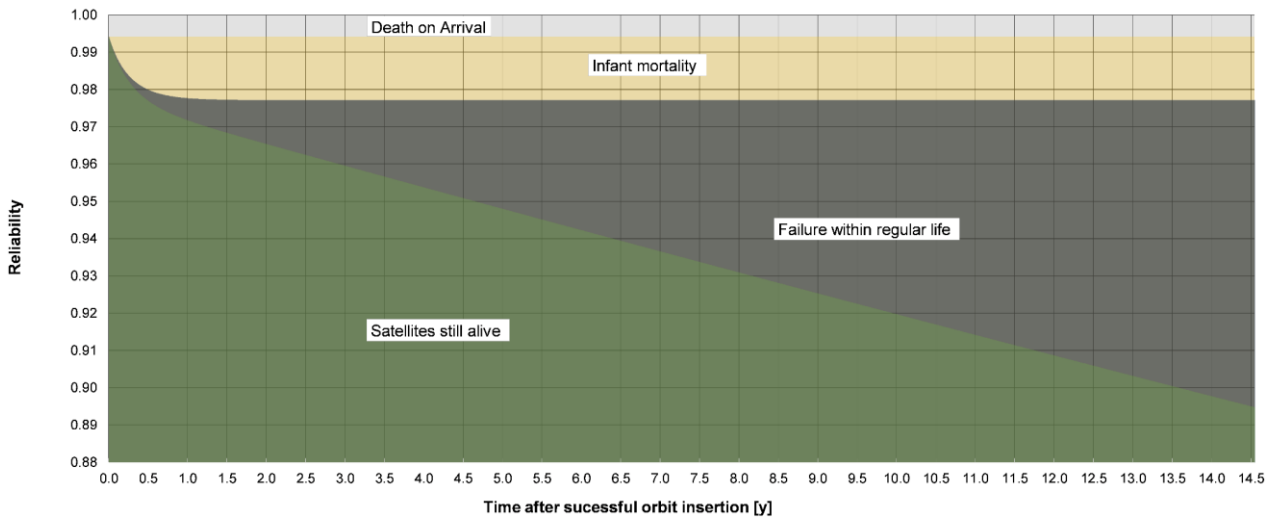


Figure 4-22: Fraction of satellites failed due to the three different portions of the PNZ modified 2-Weibull mixture function (equation (33)) within 14.5 years. Green depicts satellites still alive, yellow satellites failed due to the infant mortality portion ($\beta_1 = 0.9342$) of the function, black satellites failed due to the constant failure rate portion ($\beta_2 = 1$) of the function, and grey satellites that were never in a functional state, thus failed at activation ($p_{NZ} = 0.9941$).

The 25th percentile (-0.30%) and the 75th percentile (0.03%) as well as the whiskers of the residuals show that the fit slightly moves to a more negative deviation from the nonparametric data than the other 2-Weibull mixture functions (Figure 4-23 right). In the Weibull-plot (Figure 4-23 left), the deviation from early satellite failures can be seen, as the p_{NZ} value describes a fraction of satellites already failed at $t = 0$. After the first four failures, the fit follows the nonparametric data and it can be argued that the first four failure points at day one, two, three and four (see Table 6-7) can be seen as failures on arrival/at activation, thus at $t = 0$.

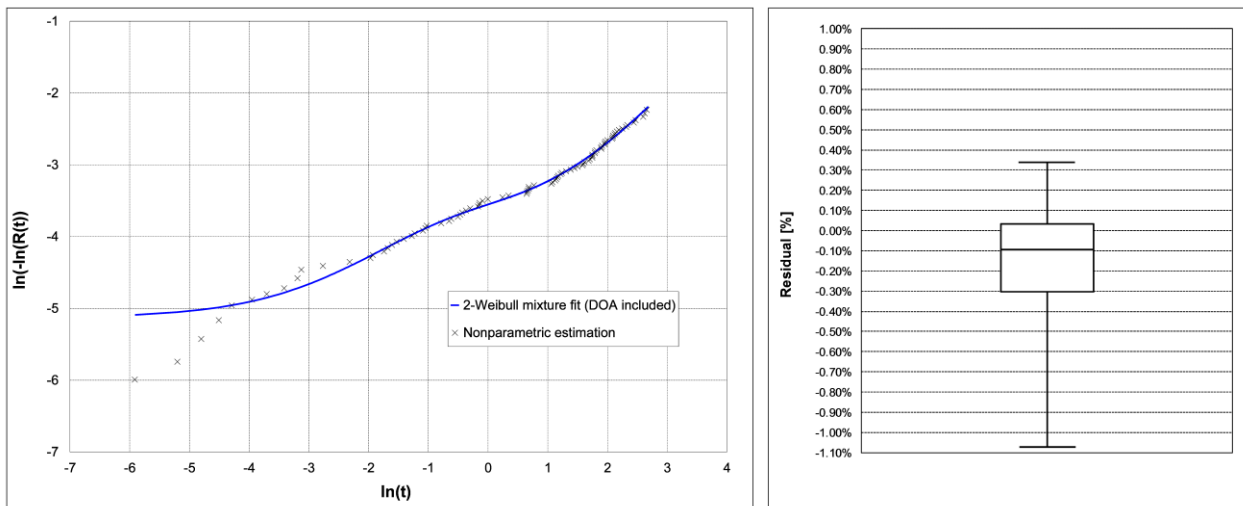


Figure 4-23: Weibull-plot (left) and boxplot (right) of the residuals between the PNZ modified 2-Weibull mixture fit (equation (33)) and the nonparametric estimation. Data source of nonparametric estimation: [115]

The difference of the PNZ-modified 2-Weibull mixture function of equation (33) to the fits of Castet & Saleh can be found in Appendix B, Figure 6-2 and Figure 6-3.

To summarize, the shown parametric functions follow the nonparametric data well and are similar to the fits presented by Castet & Saleh. However, as already pointed out, the parametric models of this dissertation take the physical plausibility of the different terms of the 2-Weibull mixture function into account. As shown, all models by Castet & Saleh have similar inexplicable behaviors regarding “late infant mortality” and “early

wear-out”, which cannot be described fully by the author of this thesis. The models presented in this work try to fit the nonparametric data with infant mortality terms that don’t persist until late in life, as this would violate the definition of infant mortality. Furthermore, our parametric functions mostly show constant or near-constant failure rates for later regions of the observation window – thus wear-out cannot be clearly distinguished from the mixed group of satellites, which is also in accordance with the presented parametric fit with a right-open bathtub-shaped failure rate function. This means that either wear-out is masked, too less to protrude in the quantity of analyzed satellites, or prevented by retirement.

In the opinion of the author, the presented 2-Weibull mixture fits, either with or without PNZ modification, describe the on-orbit behavior of the studied group of satellites to an acceptable degree. As it can be seen from different figures, all presented parametric fits show similar deviations from the nonparametric data around $t = 3$ years, $t = 5$ years, $t = 9$ years and $t = 13$ years. The assumption for that is, that the mixture of different classes of satellites led to this behavior of the nonparametric data. As argued by Levine [85], Baker & Baker [86] and by Dubos et al. [15], only satellites of the same class or family should be binned together when doing statistical analysis. The publication of Dubos et al. [15] uses the same group of satellites as the analysis of Castet & Saleh and Saleh & Castet but differentiates between certain mass classes of satellites, namely small satellites ($m < 500$ kg), medium satellites ($500 < m < 2,500$ kg) and large satellites ($m > 2,500$ kg). Since the mass of the satellite is a good reference value for other characteristics, data from Dubos et al. [15] were used to study the deviations. Figure 4-24 to Figure 4-26 show the nonparametric reliability of the mixed mass bin, with green squares for all known failures of small satellites, blue circles for all known failures of medium satellites, and red crosses for all known failures of large satellites. Steps that are not marked cannot be associated to a specific satellite class with the data provided by Dubos et al. From these data it can be recognized that the failures of small satellites are the overall dominating factor in the first month after orbit insertion. Afterwards, failures of small satellites concentrate mainly in the region between three and eight years. Medium satellites are more evenly distributed than small satellites and no clear region with dominating failures emerges. Notable, only one late failure of a medium satellite was registered beyond 8.4 years. Large satellites show 10 failures within year one, but only two of these 10 failures happen within the first month on-orbit. From the pool of satellites with known masses, the large class of satellites contributes half of all failure cases beyond year eight.

Further using graphical data of Dubos et al. [15], the nonparametric reliability estimation of the isolated mass classes was rebuilt, as can be seen in Figure 4-27. Although reduced tabular data of Saleh & Castet [22] would also have been available for that purpose, the original data of Dubos et. al [15] were used since Saleh & Castet and also Dubos et al. [118] removed satellites that failed after they reached their design lifetime from the sample, since it was not seen as desirable for this study. As can be seen in Figure 4-27, the nonparametric estimation was stopped at the point of the last failure, as done before and suggested by Saleh & Castet [22] based on publications by Kalbfleisch & Prentice. Furthermore, as presented by Dubos et al. [118], two late failures in the small and medium class of satellites were cutoff, since their impact on the overall fit of the parametric estimation is low before that point in time, but a failure anywhere in the failure-free region of more than 3.5 years (small satellites) and more than 5 years (medium satellites) would significantly impact the parametric analysis. Thus, nonparametric and parametric models were built for both cases, with and without cutoff, and the sensitivity of the overall fit of this measure was studied. While Figure 4-27 shows the aforementioned rebuilt nonparametric fit of the complete set of small, medium and large satellites, Figure 4-28 depicts the same nonparametric fit with a cutoff at $t = 10.27$ years for small satellites and $t = 8.4$ years for the medium satellites. Note that for the small class of satellites, five failures within day one and nine were merged to day one, as this represents the dead-on-activation failure class. For the other classes of satellites, only one (medium class) and two (large class) exist in this timeframe and are thus handled separately, as can be seen in the nonparametric estimation.

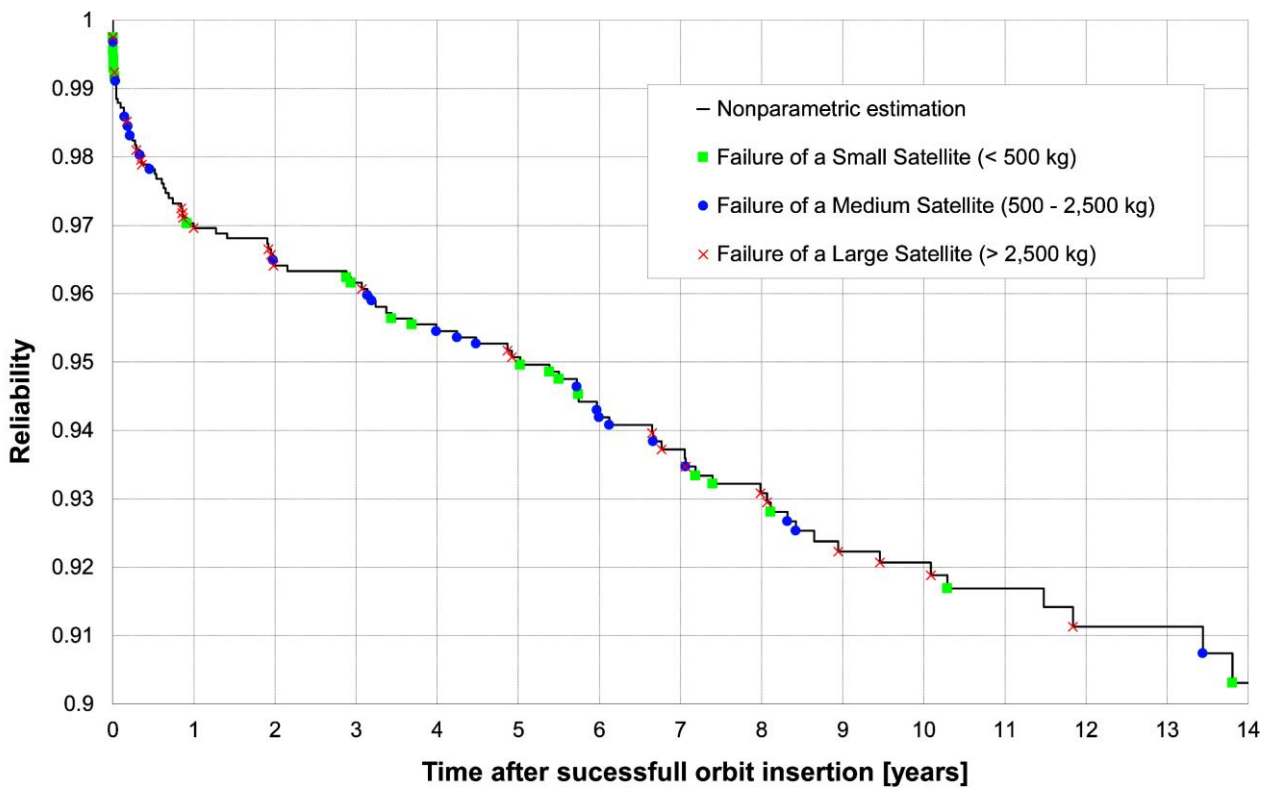


Figure 4-24: Failures of small (green square), medium (blue circle) and large satellites (red cross) marked in the complete nonparametric reliability estimation fit of satellite reliability. The mass of satellites of not marked failures is not known. Data estimated based on figures from Dubos et al. [15].

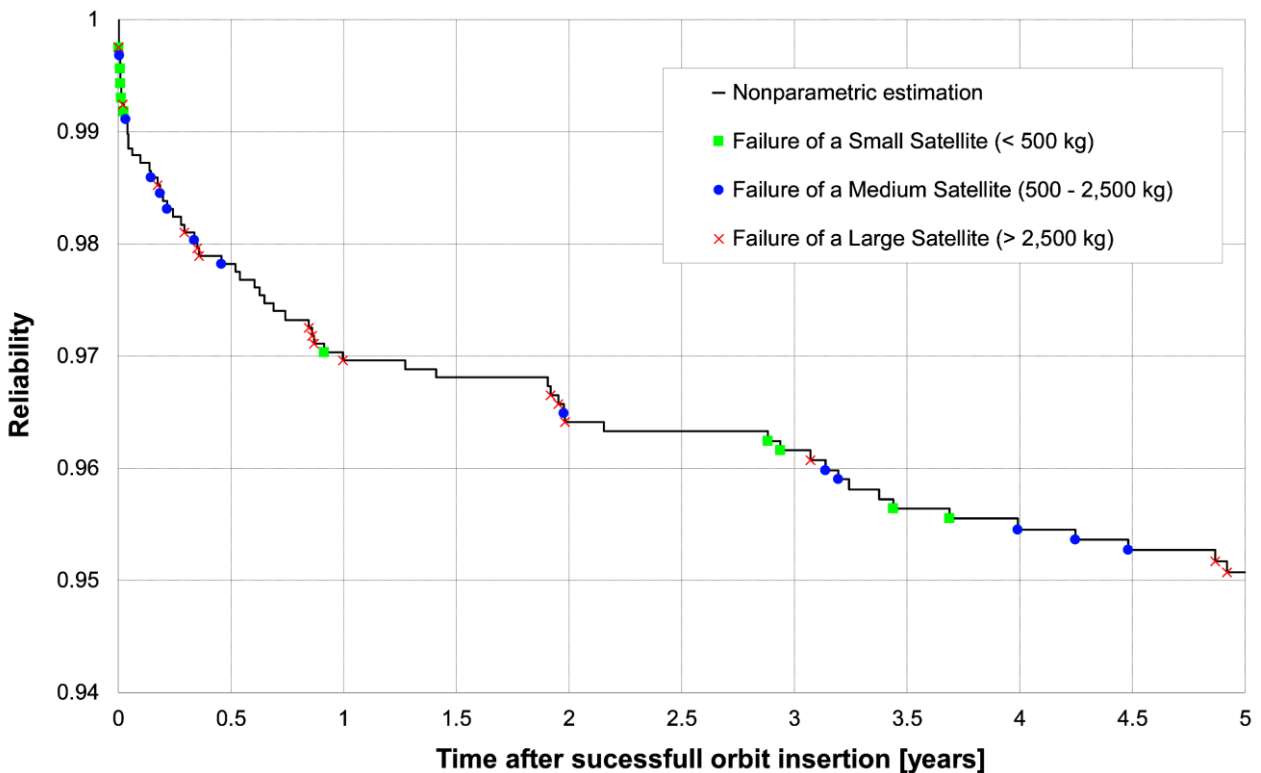


Figure 4-25: Failures of small (green square), medium (blue circle) and large satellites (red cross) marked in the nonparametric reliability estimation fit of satellite reliability (observation window reduced to one year). Data estimated based on figures from Dubos et al. [15].

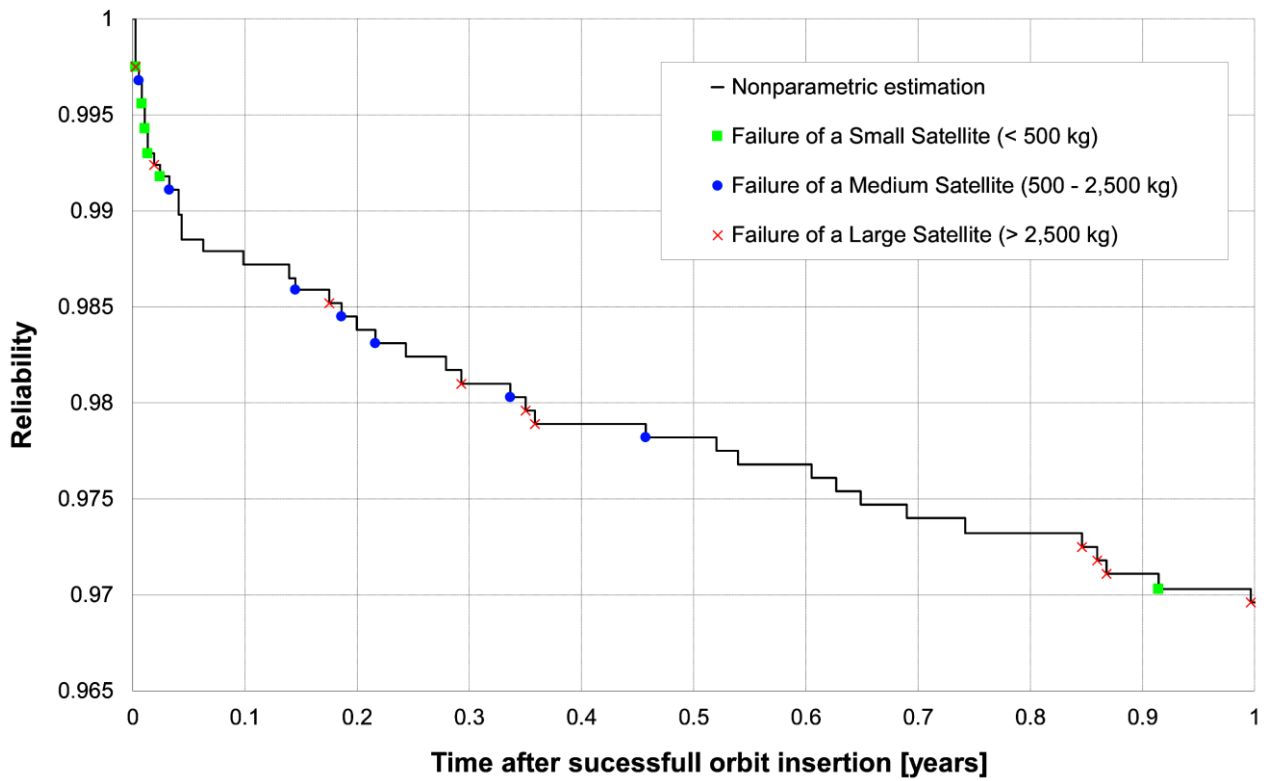


Figure 4-26: Failures of small (green square), medium (blue circle) and large satellites (red cross) marked in the nonparametric reliability estimation fit of satellite reliability (observation window reduced to one year). Data estimated based on figures from Dubos et al. [15].

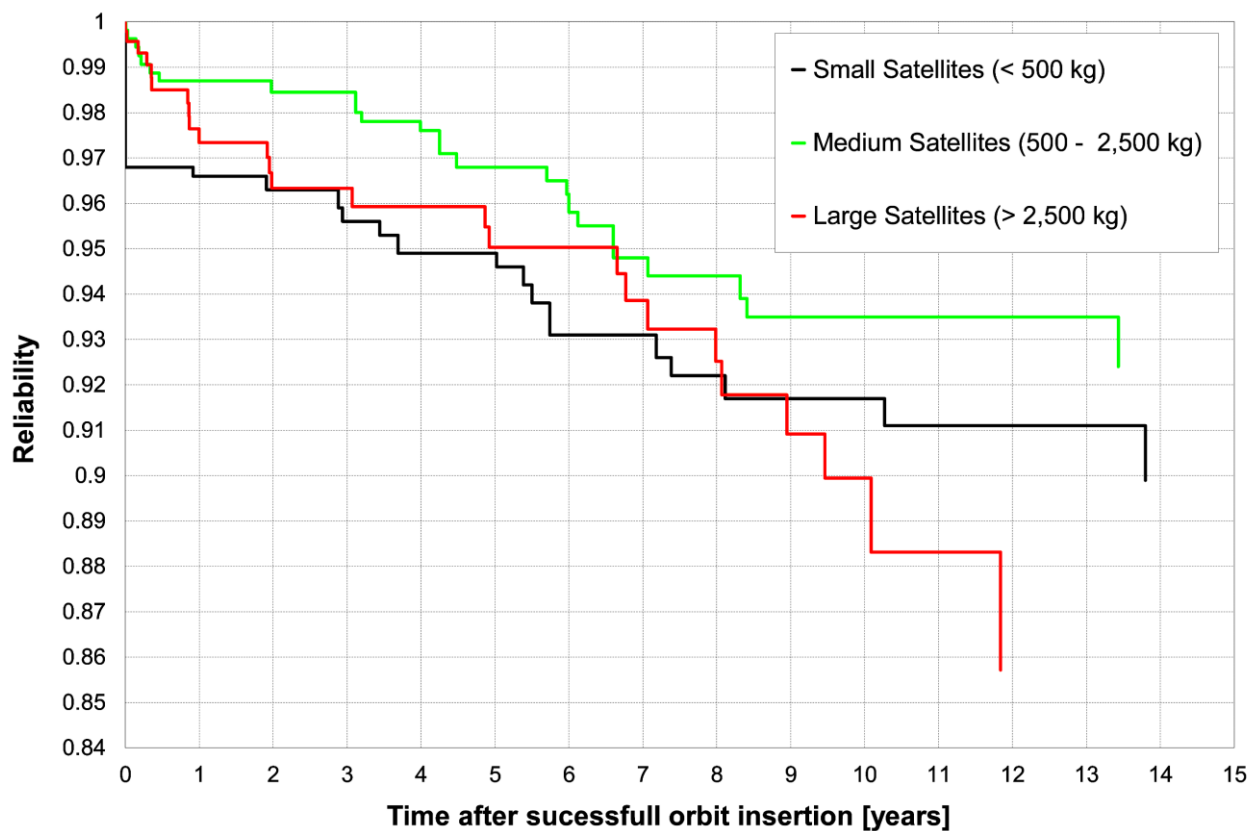


Figure 4-27: Nonparametric estimation of the complete set of small, medium and large satellites. Rebuilt based on figures from Dubos et al. [15].

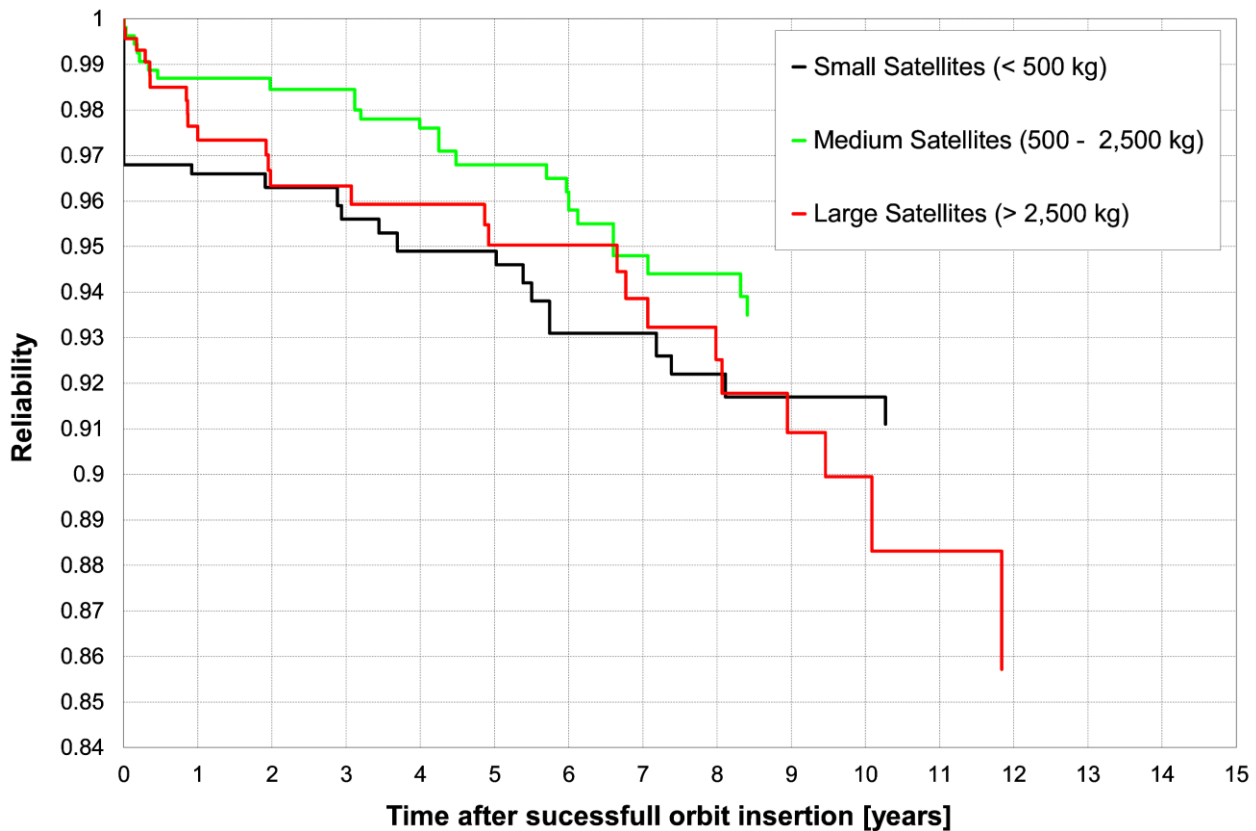


Figure 4-28: Nonparametric estimation of the set of small, medium and large satellites with cut-off at $t = 10.27$ years for small satellites and $t = 8.4$ years for medium satellites. Rebuilt based on figures from Dubos et al. [15].

In the following, we will analyze the parametric fits found by Dubos et al. [15] for the different mass classes and subsequently present new parametric fits, starting with the class of small satellites.

4.1.1 Analysis of Small Satellite Reliability

For small satellites, Dubos et al. [15] found a 2-Weibull mixture function (see equation (34)) with a very high scale factor (10^7 years) for the infant mortality portion, similar as in the work of Castet & Saleh. Also, it can be noted that the second term describes a wear-out function with a shape factor of 2.754 but again uses a relatively small scale factor of 7.3 years. Figure 4-29 shows the fraction of satellites failed due to the two different terms described by the following reliability function:

$$R(t) = 0.9607 \cdot \exp \left[- \left(\frac{t [y]}{10^7} \right)^{0.2101} \right] + 0.0393 \cdot \exp \left[- \left(\frac{t [y]}{7.3} \right)^{2.754} \right] \quad (34)$$

It can be noticed that the infant mortality portion of the function continuously grows until the end of the observation window. Due to the moderate scale factor, the wear-out rate increases until approximately $t = 7$ years, and decreases after that. At $t = 7$ years, 4.8% of reliability reduction will be due to the infant mortality term, while 2.3% caused by the wear-out term. At the end of the observation window, these values increase to 5.5% (infant mortality) and 3.9% (wear-out). Overall, the reliability dropped approximately by 9.5% at the end of the observation window.

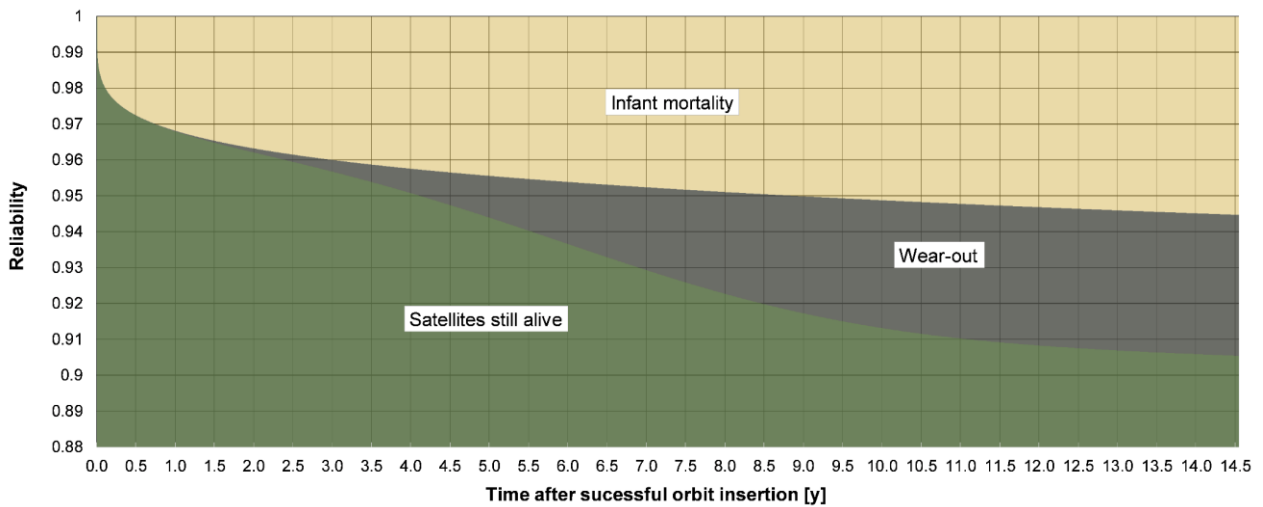


Figure 4-29: Fraction of satellites failed due to the two different portions of the 2-Weibull mixture function of Dubos et al. [15] (equation (34)) within the first 14.5 years on-orbit. Green depicts satellite still alive, yellow satellites failed due to the infant mortality portion ($\beta_1 = 0.2101$) of the function, and black satellites failed due to the wear-out portion ($\beta_2 = 2.754$) of the function.

Figure 4-30 shows the fraction of satellites that failed due to the infant mortality term over the whole observation window. Again, although the term increases quickly, as one would expect, the continued growth throughout the observation window cannot be explained by the author of this thesis. Figure 4-31 shows the same for the wear-out portion of the parametric fit. As noted before, the wear-out growth rate increases up to a point of approximately $t = 7$ years, and decreases afterwards. Almost no wear-out can be noticed late in the lifetime, contradicting the expectations based on the physical idea behind wear-out.

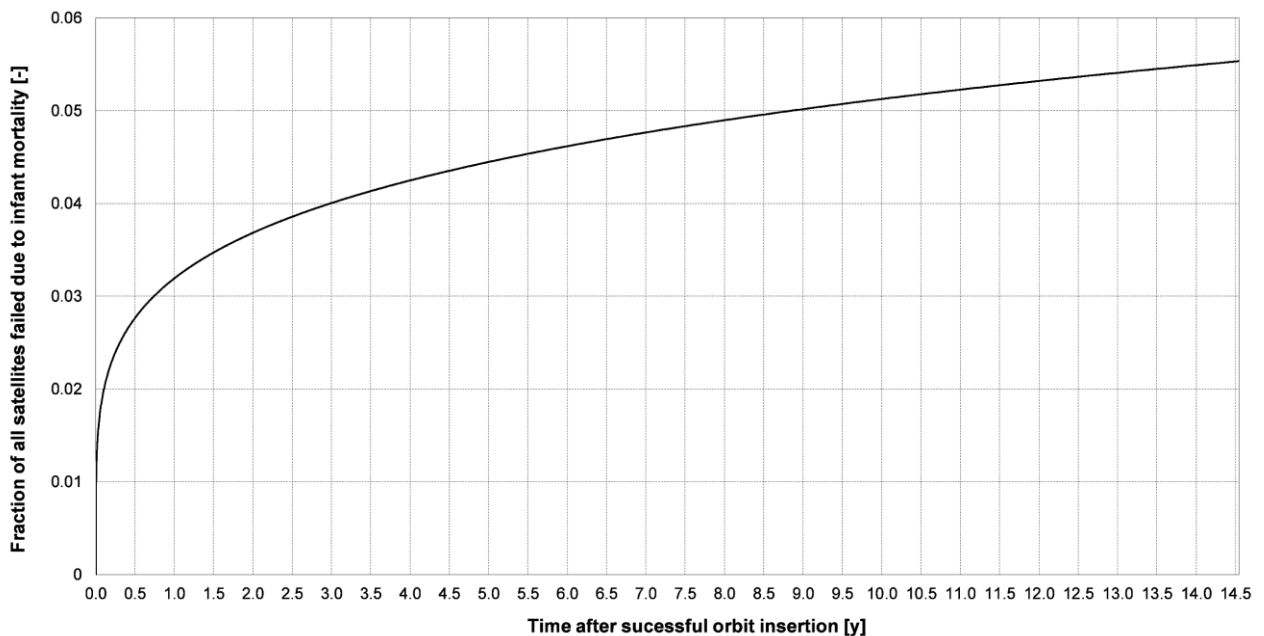


Figure 4-30: Fraction of all satellites failed due to the infant mortality term of the 2-Weibull mixture function of Dubos et al. [15] (equation (34)).

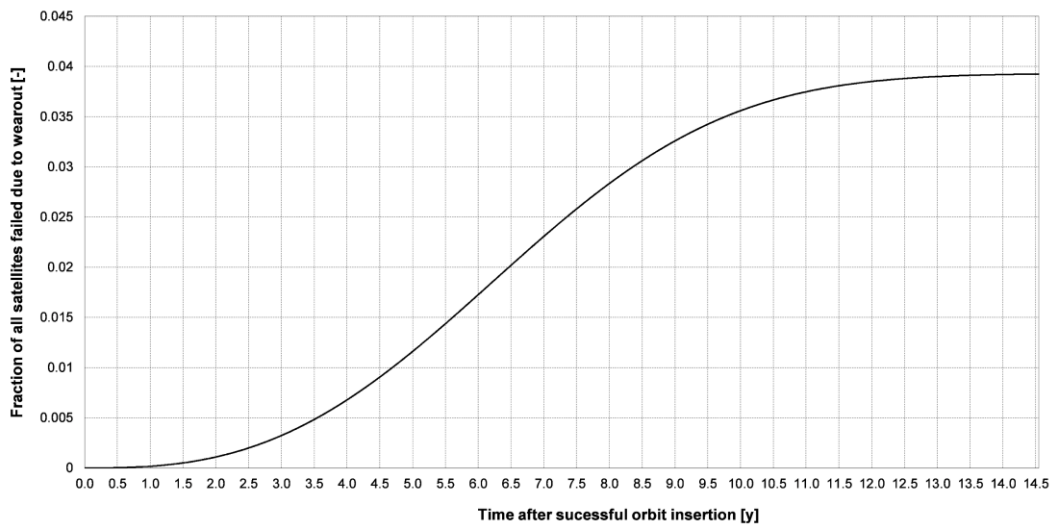


Figure 4-31: Fraction of all satellites failed due to the wear-out term of the 2-Weibull mixture function of Dubos et al. [15] (equation (34)).

Based on the nonparametric data presented in Figure 4-27, a parametric fit of the full dataset of small satellites was made. Since this class of satellites showed clear signs of dead on arrival/activation, a Single-Weibull function was modified by a p_{NZ} factor, this time fixed to a value of 0.968 and the shape factor set to a value of one. Thus, the nonlinear least squares fitted Single-Weibull function is:

$$R(t) = 0.968 \cdot \exp \left[- \left(\frac{t [y]}{174.5} \right)^1 \right] \quad (35)$$

The resulting parametric fit has a goodness-of-fit value of $R^2 = 0.9525$, and the shape factor of the Weibull function denotes a fraction of satellites that failed with constant failure rate ($\beta = 1$). The scale factor of the function is relatively high with $\theta = 174.5$ years (95% confidence interval: 83.04 years, 266.1 years). The resulting fit can be seen in Figure 4-32. The constant failure rate leads to a near-linear parametric fit within the observation window, which starts at $t = 0$ with $R = 0.968$ due to satellites failing directly on activation, which is considered by the p_{NZ} modifier.

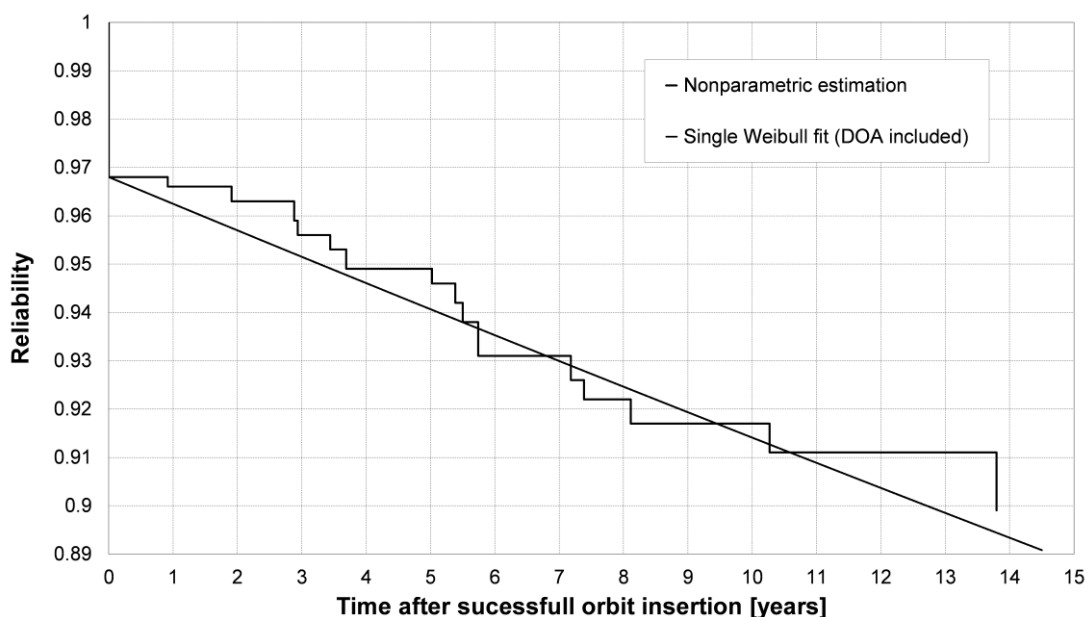


Figure 4-32: PNZ modified Single-Weibull fit (equation (35)) of small satellite reliability. Data source of nonparametric estimation: [15]

Looking at the fractions of the satellites failed due to the two different terms of the Single-Weibull fit, one can recognize the relatively large impact of early failures on the overall reliability. Overall, the class of small satellites reaches a reliability of 89.1% at the end of observation. Of a total reduction of 10.9%, 3.2% stems from satellites that were characterized as dead on arrival/activation.

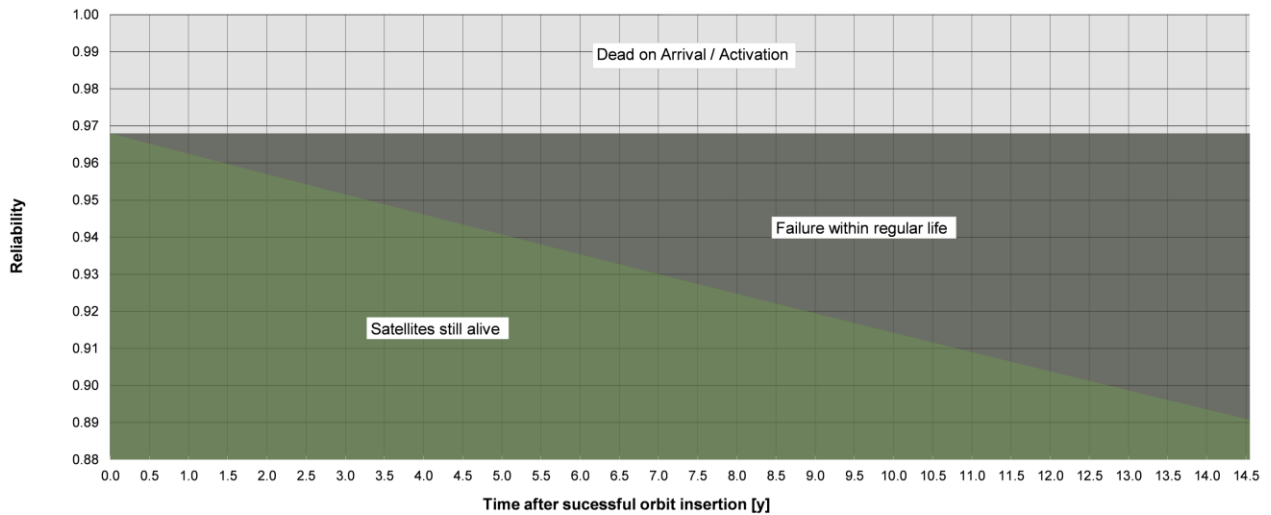


Figure 4-33: Fraction of small satellites failed due to the two different portions of the Single-Weibull function (equation (35)) within the first 14.5 years on-orbit. Green depicts satellite still alive, white satellites failed due to dead on arrival/activation cases ($1 - p_{NZ} = 0.032$), and black satellites failed due to the constant failure rate portion ($\beta = 1$) of the function.

The Weibull-plot (Figure 4-34 left) shows that the alignment between the nonparametric and the parametric model is already good. Within the boxplot, the 25th percentile (-0.68%) and the 75th percentile (0.005%) as well as the long whiskers show that there is still room for improvement (Figure 4-34 right). The outliers are stemming from nonparametric data points at day zero and one, as all failures between day one and nine were bundled in day one in the nonparametric estimation.

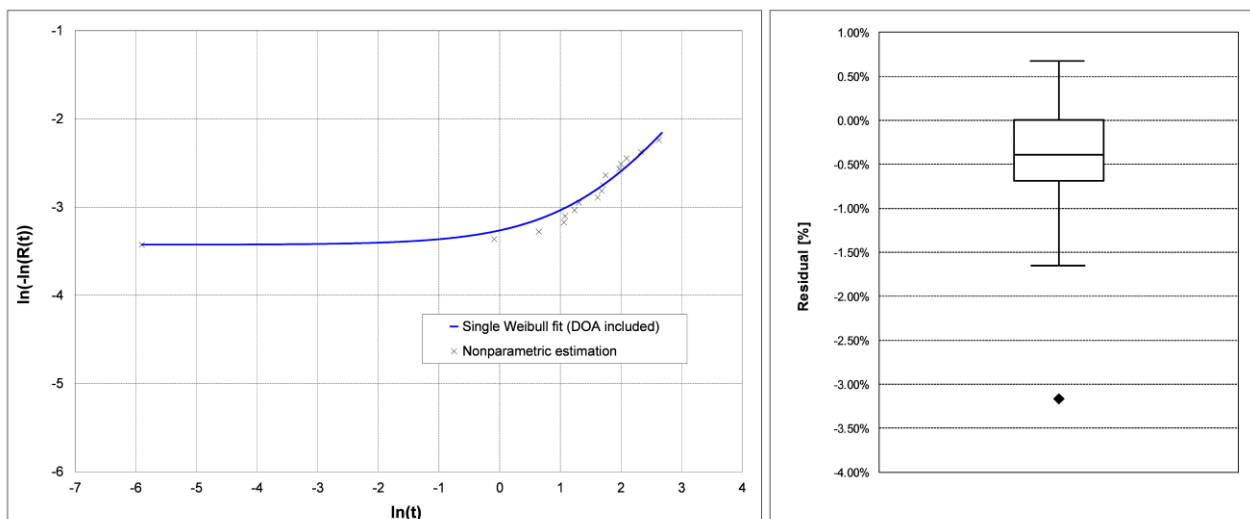


Figure 4-34: Weibull-plot (left) and boxplot (right) of the residuals between the PNZ modified Single-Weibull fit (equation (35)) and the nonparametric estimation of small satellite reliability data. Source of nonparametric estimation: [15]

The deviation from the original fit by Dubos et al. as well as the fraction of satellites that failed due to the constant failure rate term of the function can be found in Appendix B, Figure 6-4 to Figure 6-6.

As a next step, the reduced dataset up to $t = 10.27$ years was fitted with a PNZ-modified 2-Weibull mixture function:

$$R(t) = 0.968 \cdot \left(0.0273 \cdot \exp \left[- \left(\frac{t [y]}{64.59} \right)^{0.9759} \right] + 0.9727 \cdot \exp \left[- \left(\frac{t [y]}{86.98} \right)^{1.283} \right] \right) \quad (36)$$

The resulting parametric function (see Figure 4-35) has a goodness-of-fit value of $R^2 = 0.9658$. The function is dominated by the second term, which has a shape factor slightly above the constant failure rate. Since also the first term has a shape factor near the constant failure rate, it seems superfluous to use a 2-Weibull mixture model on the reduced dataset, since the cases of very early failure are already mostly covered by the p_{NZ} factor.

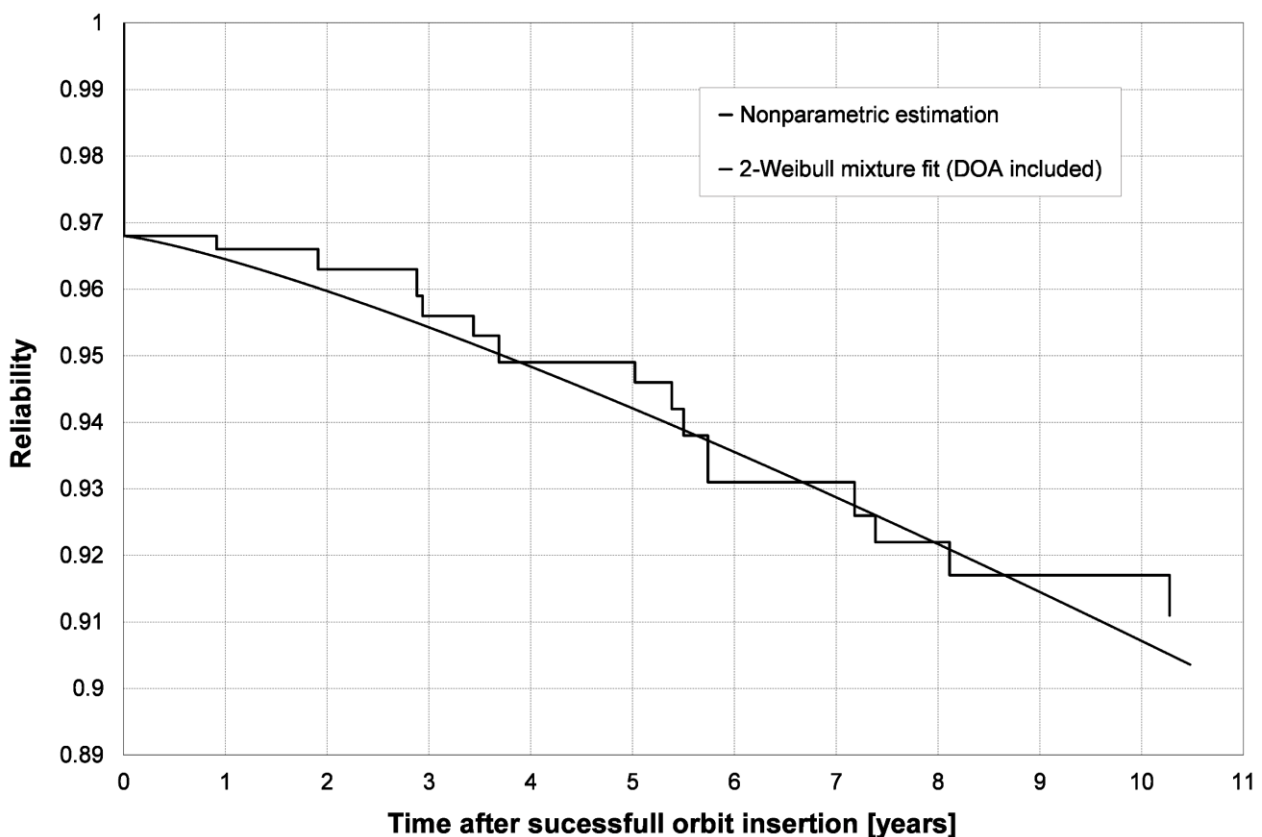


Figure 4-35: PNZ modified 2-Weibull mixture fit (equation (36)) of small satellite reliability of a dataset that was cut-off at $t = 10.27$ years. Data source of nonparametric estimation: [15]

Fitting a PNZ modified Single-Weibull to the reduced dataset, a similar goodness-of-fit ($R^2 = 0.9661$) can be obtained by a function with less parameters (see Figure 4-36):

$$R(t) = 0.968 \cdot \exp \left[- \left(\frac{t [y]}{87.2} \right)^{1.262} \right] \quad (37)$$

The shape factor of the function shows a slightly increasing failure rate and also has its 95% confidence interval above a value of one (1.064, 1.461). The scale factor is $\theta = 87.2$ years, with a 95% confidence interval of 52.59 years to 121.8 years. As depicted in Figure 4-37, the overall fraction of satellites that failed very early is the same magnitude as in the fits before. The slightly increasing failure rate, although technically a sign of wear-out, is denoted as failures within the regular life of the small satellites, as it is influencing the reliability of small satellites over the whole observation window. The proximity to the constant failure rate

could mean that a mix of different mechanisms is responsible for the bundled on-orbit reliability of small satellites. As it can be seen from Figure 4-36, the staircase function as well as the parametric fit do not show any sign of more pronounced wear-out at later stages on-orbit.

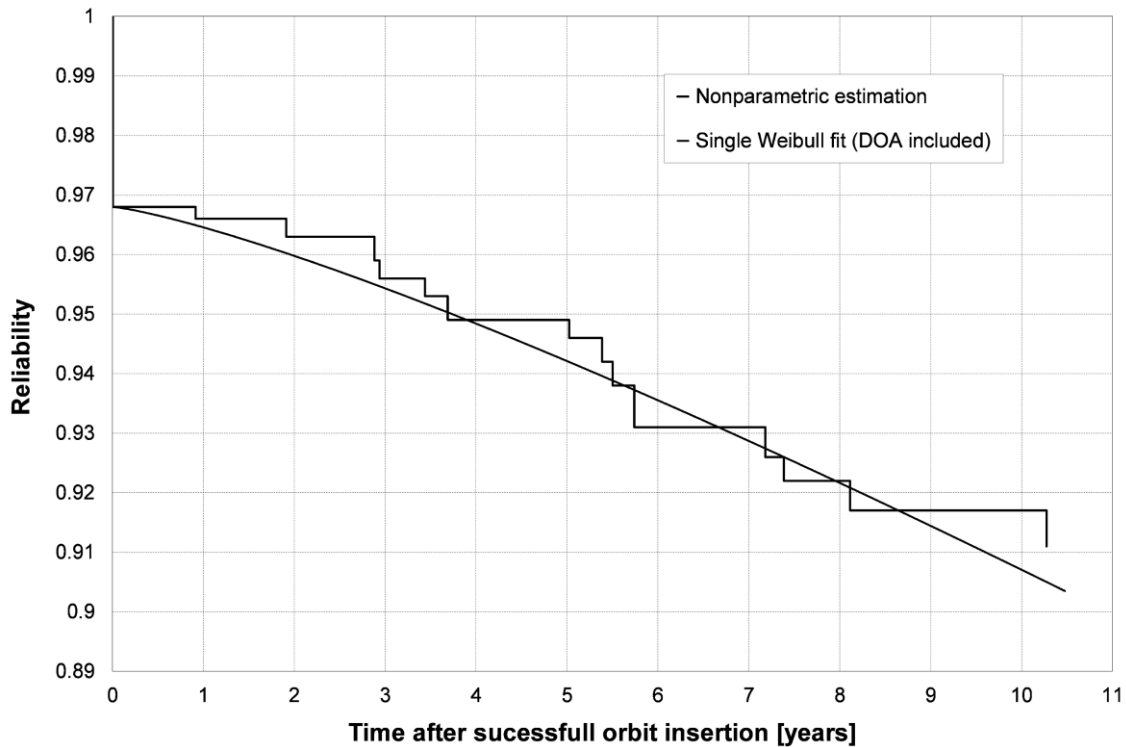


Figure 4-36: PNZ modified Single-Weibull mixture fit (equation (37)) of small satellite reliability of a dataset that was cut-off at $t = 10.27$ years. Data source of nonparametric estimation: [15].

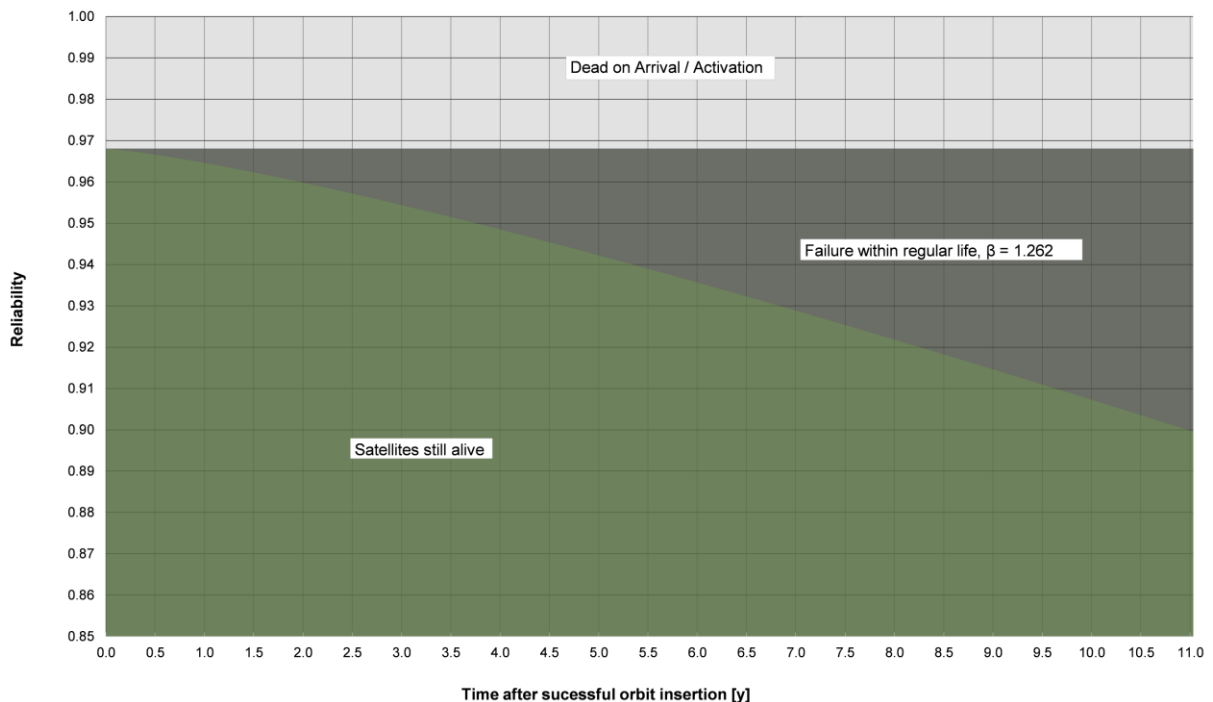


Figure 4-37: Fraction of small satellites failed due to the two different portions of the Single-Weibull function (equation (37)) within the first 11 years on-orbit. Green depicts satellite still alive, white satellites failed due to dead on arrival/activation cases ($1 - p_{NZ} = 0.032$), and black satellites failed due to the increasing failure rate portion ($\beta = 1.262$) of the function.

At the end of the observation window, the overall reliability is reduced by 10% of which 6.8% stem from the term with the increasing failure rate (see also Figure 4-38).

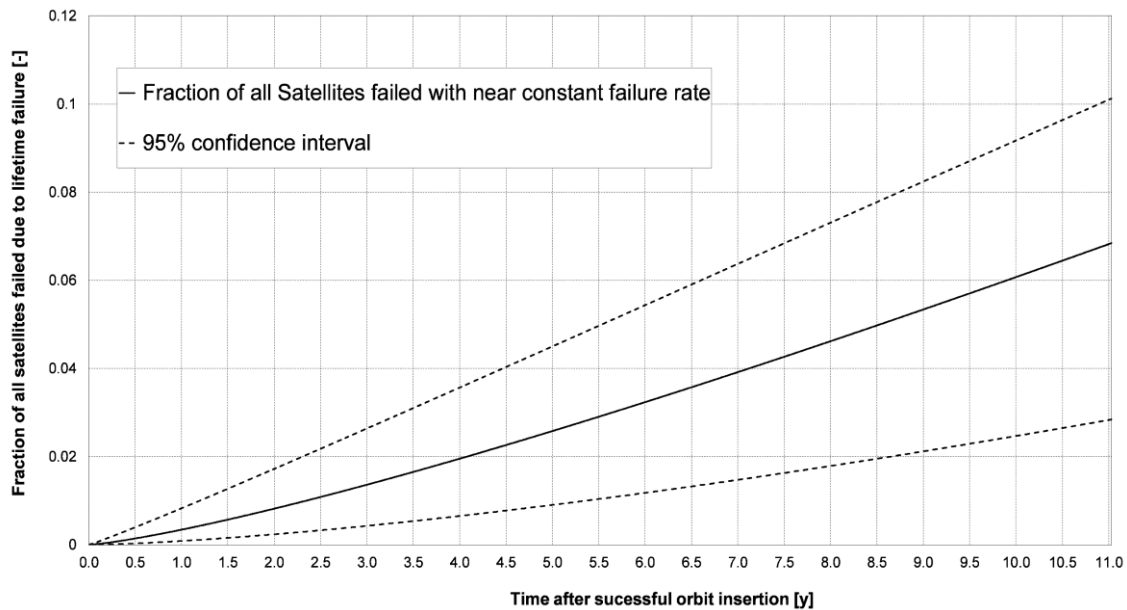


Figure 4-38: Fraction of all satellites failed due to the increasing failure rate term of the PNZ modified Single-Weibull function (equation (37)).

The Weibull-plot (Figure 4-39 left) as well as the 25th percentile (-0.46%) and the 75th percentile (0.004%) of the boxplot (Figure 4-39 right) show a better alignment between the nonparametric and the parametric model than in the full dataset. As before, the outliers are stemming from nonparametric data points at day zero and one.

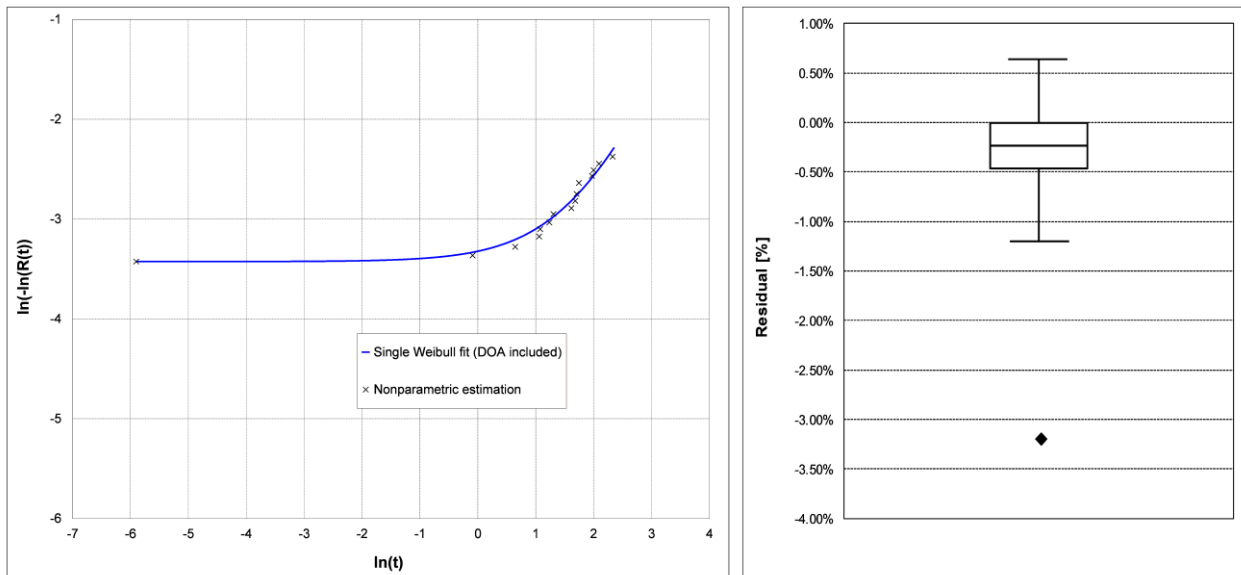


Figure 4-39: Weibull-plot (left) and boxplot (right) of the residuals between the PNZ modified Single-Weibull fit (equation (37)) and the reduced nonparametric estimation of small satellite reliability data. Source of nonparametric estimation: [15]

As shown in Figure 4-40, the PNZ modified Single-Weibull fit shows a very good alignment with the before presented, PNZ-modified 2-Weibull mixture fit. Thus, the assumed superfluity of the 2 additional parameters is proven.

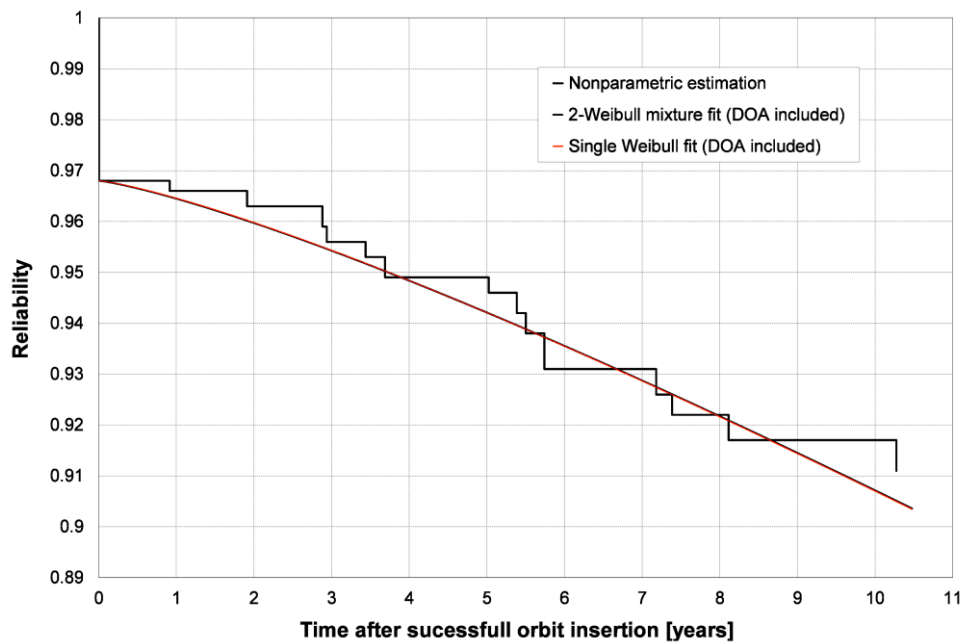


Figure 4-40: Comparison between the PNZ modified 2-Weibull mixture fit (equation (36)) and the PNZ modified Single-Weibull fit (equation (37)) of the reduced nonparametric reliability data of small satellites. Source of nonparametric estimation: [15].

The new parametric function mostly deviates less than 0.5% from the original fit by Dubos et al., as can be seen in Figure 4-41 and Figure 4-42. The larger deviations in the beginning stem from the newly implemented rate of dead on arrival/activation. After an observation window of about 10 years, the fit again deviates from the nonparametric data as well as from the original fit by Dubos et al. Although this is not ideal, it can be argued that this region at the end of the observation window has to be handled with care in any case, due to the presumably reduced number of active satellites and satellite failures in this class of satellites. Furthermore, the region beyond the second last failure of $t = 8.11$ years could also be seen as another target for data cut-off, as the distance to the last failure is more than 2 years.

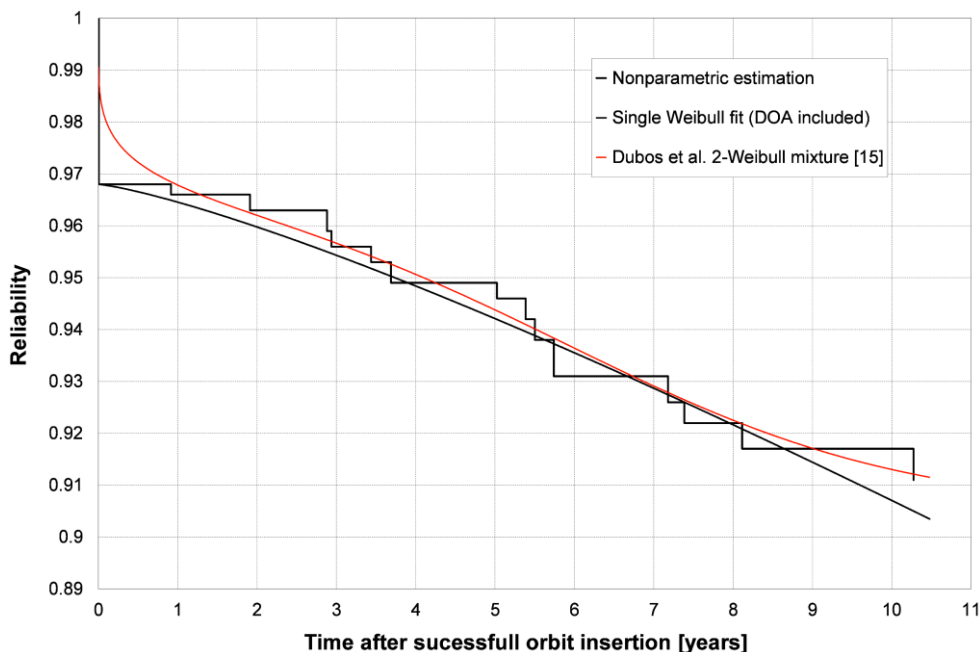


Figure 4-41: Comparison between PNZ modified Single-Weibull fit (equation (37)) of the reduced nonparametric reliability data and the 2-Weibull mixture fit by Dubos et al. [15] (equation (34)). Source of nonparametric estimation: [15]

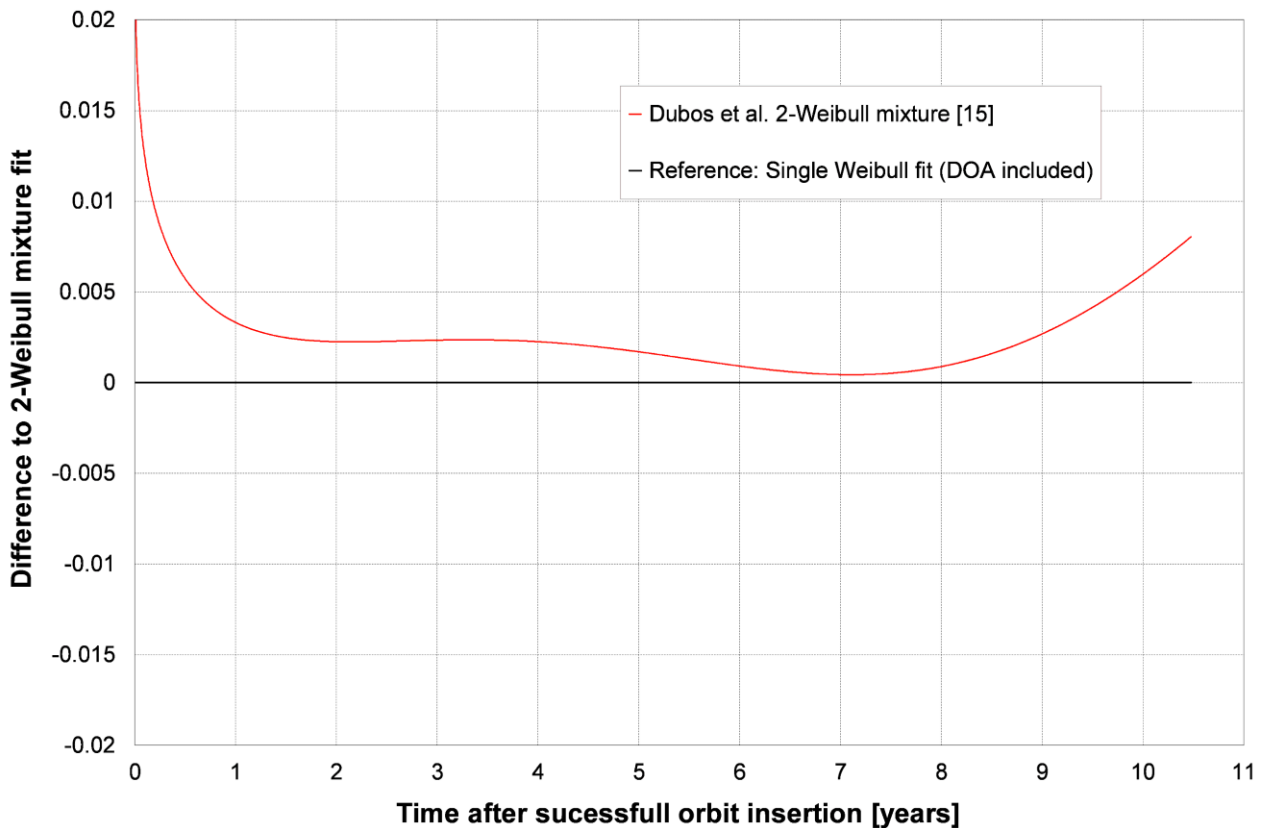


Figure 4-42: Difference of the 2-Weibull mixture model of Dubos et al. [15] (equation (34)) to the PNZ-modified Single-Weibull fit (equation (37)).

To conclude the analysis of the group of small satellites, the new found PNZ-modified Single-Weibull fit shows sufficient accuracy to the nonparametric estimation, while providing a more realistic physical explanation of the underlying parameters of the Weibull function. The data will be further discussed in Chapter 5. We will continue with the second class of satellites, namely medium satellites.

4.1.2 Analysis of Medium Satellite Reliability

For medium satellites, Dubos et al. [15] found a 2-Weibull mixture function with similar characteristics to their parametric model for small satellites:

$$R(t) = 0.9703 \cdot \exp \left[- \left(\frac{t \text{ [y]}}{6,840} \right)^{0.5071} \right] + 0.0297 \cdot \exp \left[- \left(\frac{t \text{ [y]}}{6.6} \right)^{5.538} \right] \quad (38)$$

In their parametric fit, the infant mortality portion of the function ($\beta_1 = 0.5071$) is the dominant term for which again a large scale factor ($\theta_1 = 6,840$ years) is used. The wear-out part of the function is comprised of a large shape factor ($\beta_2 = 5.538$) but a moderate scale factor ($\theta_2 = 6.6$ years). Figure 4-43 depicts the two different fractions of the parametric function within the observation window.

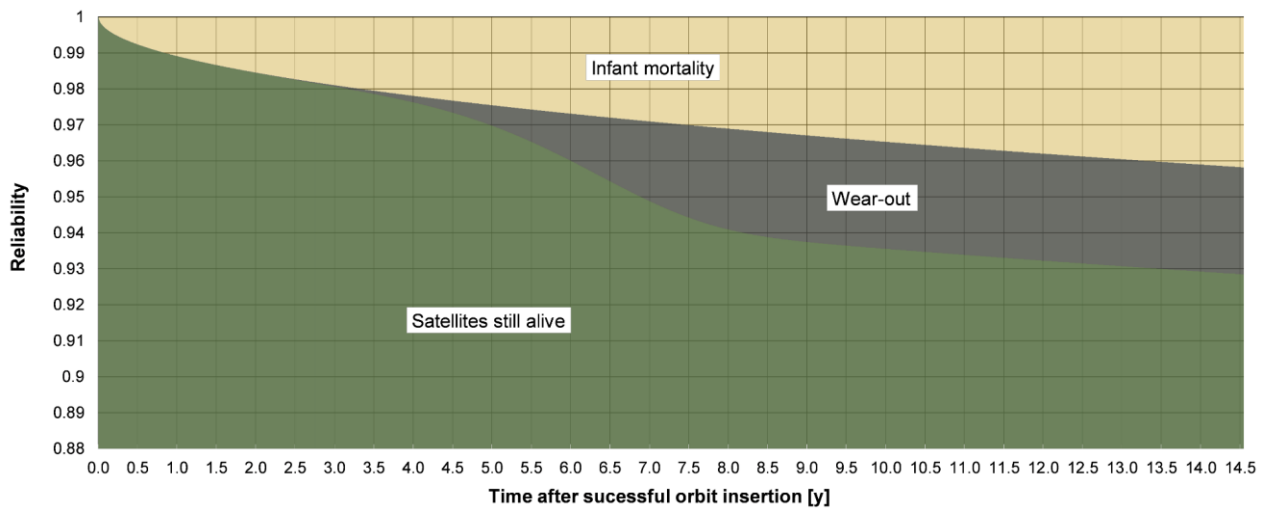


Figure 4-43: Fraction of medium satellites failed due to the two different portions of the 2-Weibull mixture function of Dubos et al. [15] (equation (38)) within the first 14.5 years on-orbit. Green depicts satellites still alive, yellow satellites failed due to the infant mortality portion ($\beta_1 = 0.5071$) of the function, and black satellites failed due to the wear-out portion ($\beta_2 = 5.538$) of the function.

As expected due to the large scale factor, the infant mortality portion of the function grows throughout the whole observation window (see also Figure 4-44). At $t = 7$ years, the overall reliability was reduced by 2.9% by the infant mortality term. At the end of the observation window, this value grows to 4.2%. Figure 4-45 shows that the wear-out portion of the function saturates at 9 years, so after this point in time only the infant mortality portion contributes to the further decline of the reliability of medium satellites. Overall, the reliability decreases to 92.8% at the end of the observation window.

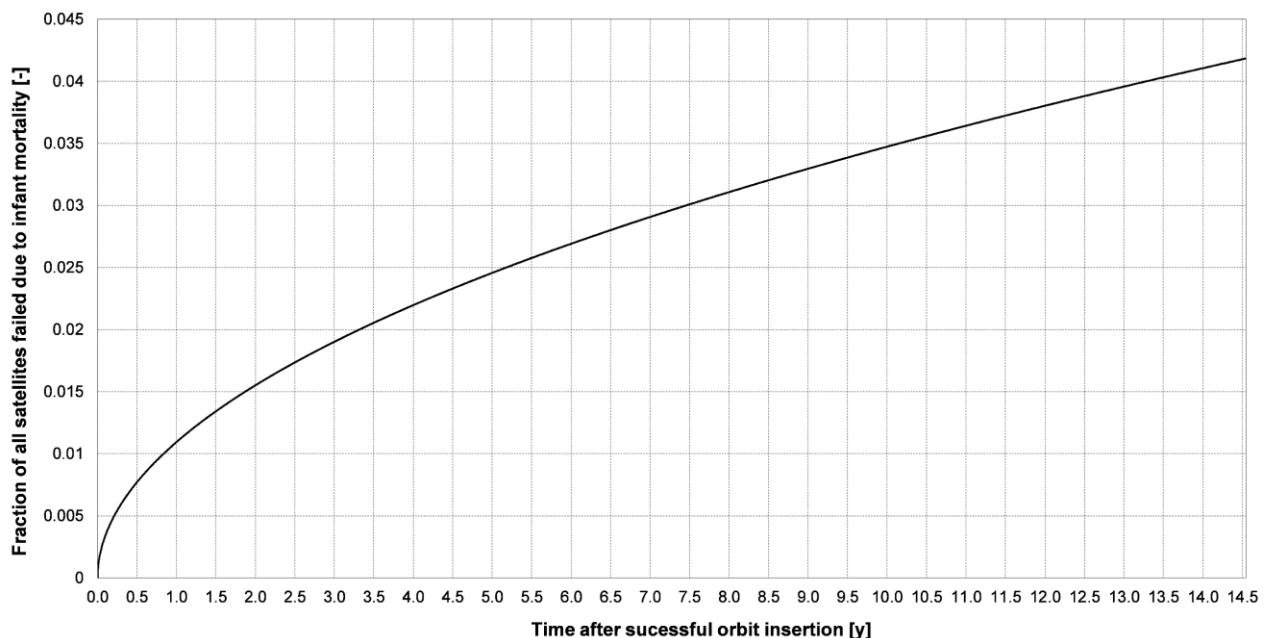


Figure 4-44: Fraction of all medium satellites failed due to the infant mortality term of the 2-Weibull mixture function of Dubos et al. [15] (equation (38)).

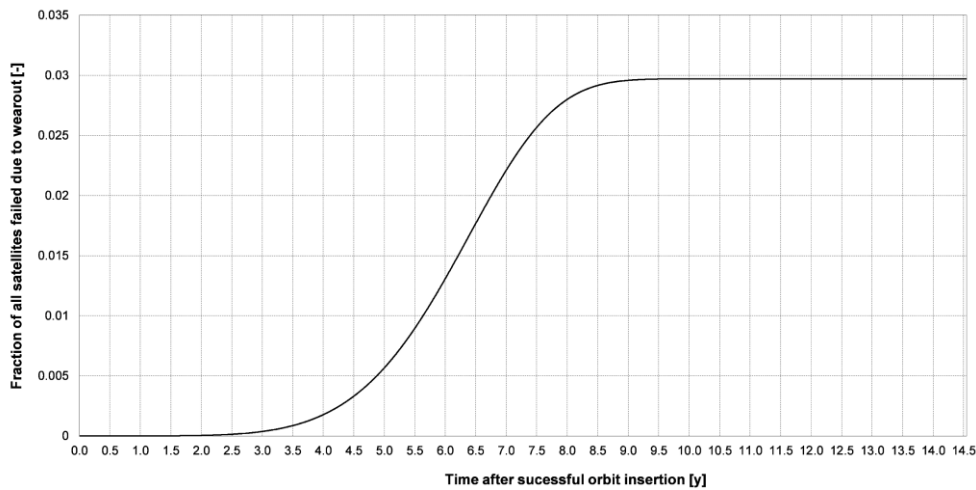


Figure 4-45: Fraction of all medium satellites failed due to the wear-out term of the 2-Weibull mixture function of Dubos et al. [15] (equation (38)).

As before, this continuous growth of infant mortality and sharp rise and decline of wear-out cannot be fully explained by the author of this thesis. Thus, the nonparametric data were used to build a new parametric fit for this group of satellites. Additionally, to the full set of nonparametric data, a reduced dataset up to a cutoff of $t = 8.4$ years, as already pointed out in Section 4.1, was used for this study and will be presented in the following. As a first result, the following 2-Weibull mixture function was found for medium satellites:

$$R(t) = 0.01128 \cdot \exp \left[- \left(\frac{t [y]}{0.1284} \right)^{0.8216} \right] + 0.98872 \cdot \exp \left[- \left(\frac{t [y]}{44.93} \right)^{1.718} \right] \quad (39)$$

Instead of a large scale factor for the infant mortality portion ($\beta_1 = 0.8216$) of the function, this fit uses a relatively benign scale factor of $\theta_1 = 0.1284$ years. The second part of the function is built by a wear-out term, which uses moderate shape and scale factors of $\beta_2 = 1.718$ and $\theta_2 = 44.93$ years. As it can be seen in Figure 4-46, the parametric fit follows the nonparametric data quite well. The goodness-of-fit is $R^2 = 0.995$. The initial drop due to infant mortality is succeeded by a long region of increasing failure rate.

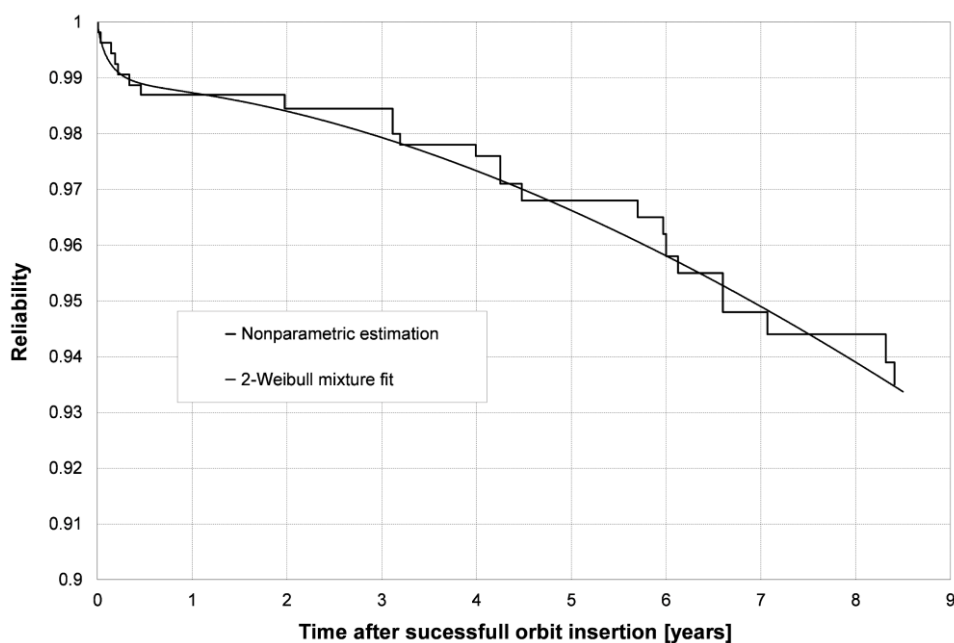


Figure 4-46: 2-Weibull mixture fit (equation (39)) of medium satellite reliability of a dataset that was cut-off at $t = 8.4$ years. Data source of nonparametric estimation: [15]

In this parametric fit, the infant mortality term flattens out after approximately one year, as depicted in Figure 4-47 and Figure 4-48. Contrary to that, the wear-out portion continues to grow until the end of the observation window (see Figure 4-49). Thus, this fit is not in conflict with the general characteristics of infant mortality and wear-out in technical products. On the negative side, the 95% confidence interval of the infant mortality term is too large, with β_1 ranging from -0.3093 to 1.953, and θ_1 from -0.1014 years to 0.3581 years. The 95% confidence intervals of the wear-out term are β_2 (1.208, 2.228) and θ_2 (24.31 years, 65.55 years).

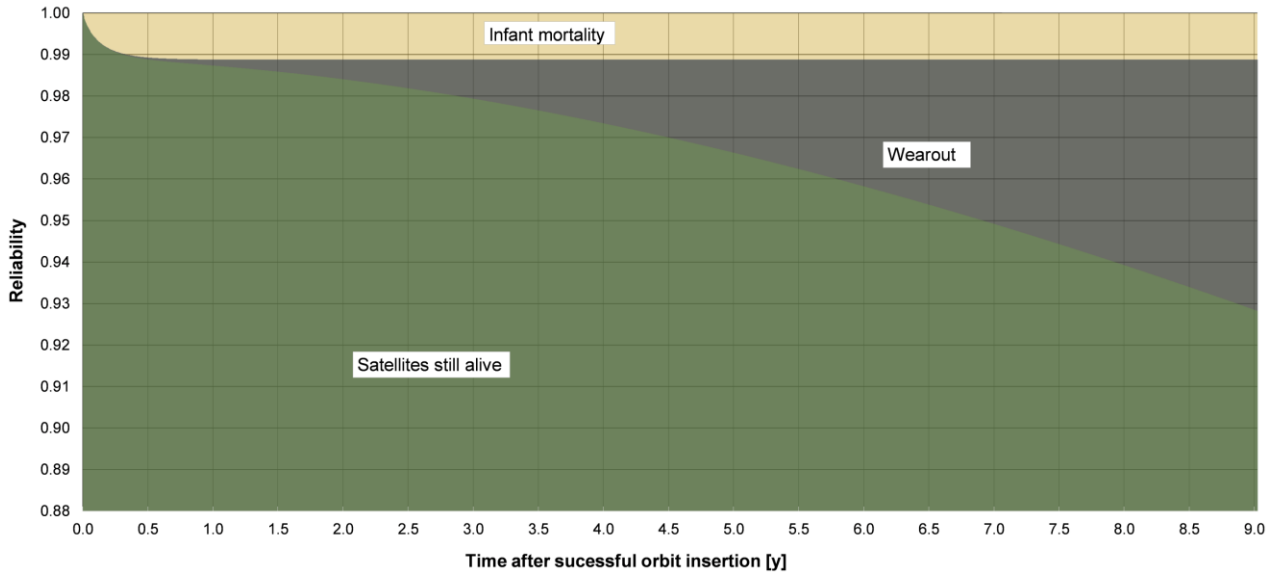


Figure 4-47: Fractions of medium satellites failed due to the two different portions of the 2-Weibull mixture function (equation (39)) within the first 9 years on-orbit. Green depicts satellites still alive, yellow satellites failed due to the infant mortality portion ($\beta_1 = 0.8216$) of the function, and black satellites failed due to the wear-out portion ($\beta_2 = 1.718$) of the function.

According to the new parametric fit, the reliability of all medium satellites will be reduced by 1.1% after one year on-orbit due to infant mortality, and only by 0.1% due to wear-out in the same timeframe. At $t = 4.5$ years the reduction contributed by worn-out satellites is grown to 1.9%, while the infant mortality fraction stays constant. At the end of the observation window, the reliability of medium satellites decreases to 92.8%. The fractions of satellites that failed due to the infant mortality portion and the wear-out portion of the parametric function can be found in Figure 4-48 and Figure 4-49.

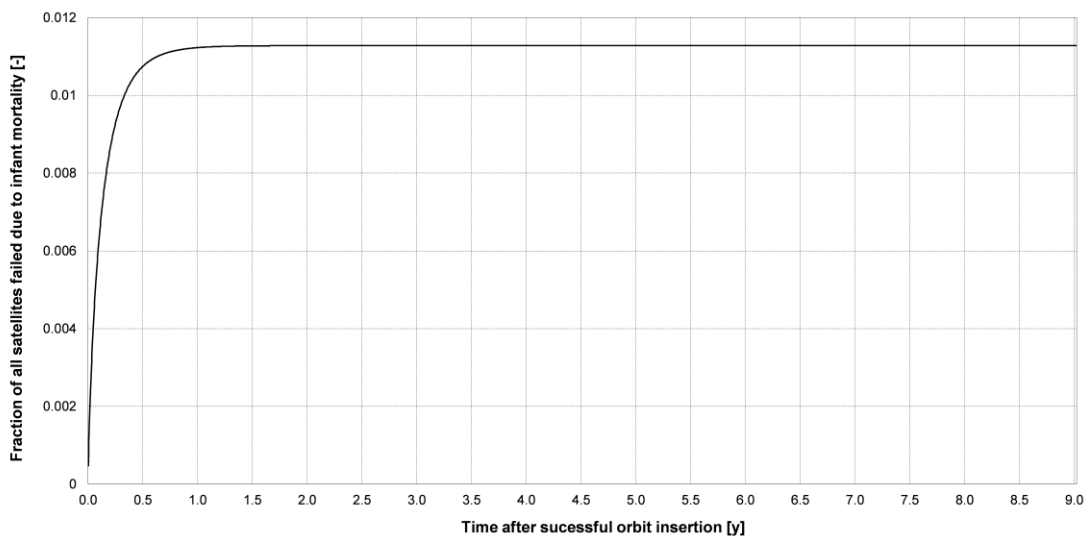


Figure 4-48: Fraction of all medium satellites failed due to the infant mortality term of the 2-Weibull mixture function (equation (39)).

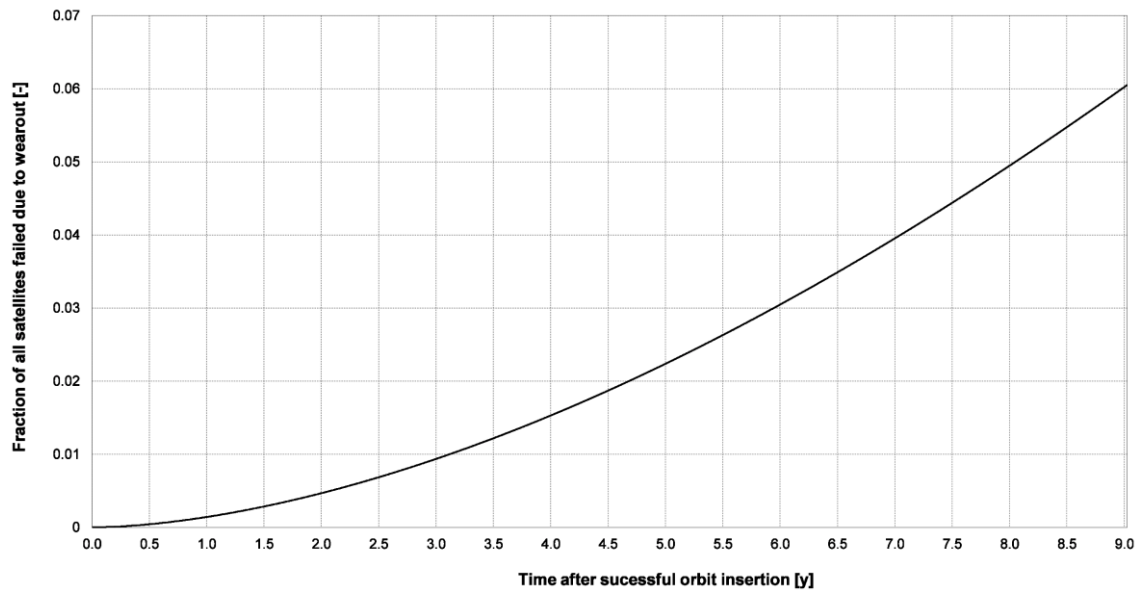


Figure 4-49: Fraction of all medium satellites failed due to the wear-out term of the 2-Weibull mixture function (equation (39)).

The Weibull-plot (Figure 4-50 left) shows a generally good alignment with the nonparametric data. Slight early deviations are overcome at later stages. The 25th percentile (-0.38%) and the 75th percentile (0.026%) of the boxplot (Figure 4-50 right) confirm that.

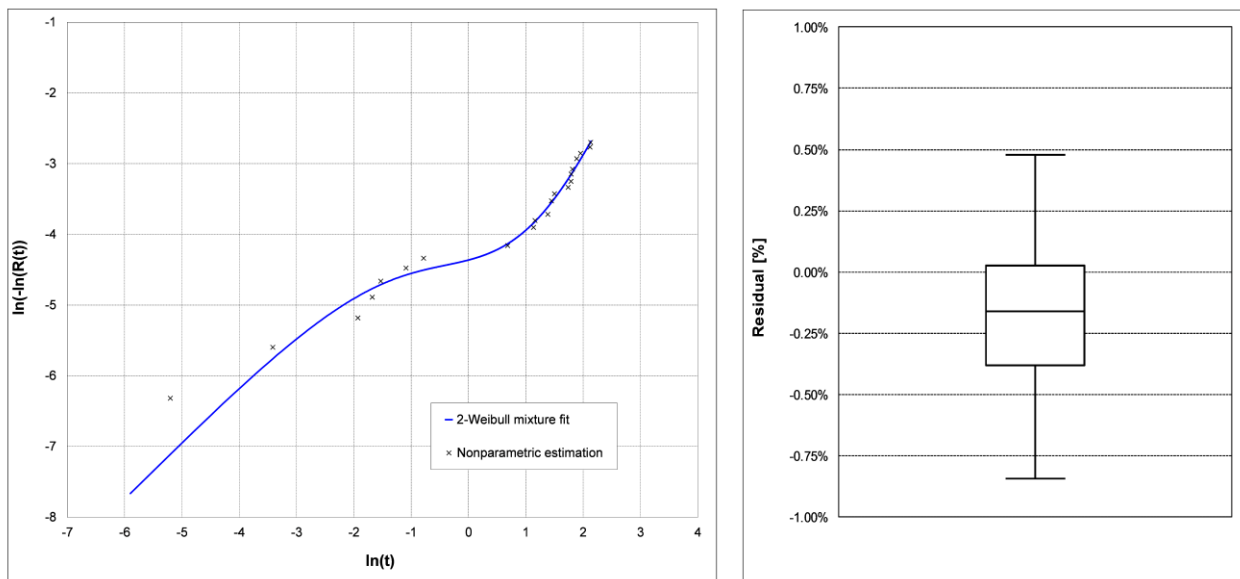


Figure 4-50: Weibull-plot (left) and boxplot (right) of the residuals between the 2-Weibull mixture fit (equation (39)) and the reduced nonparametric estimation of reliability data of medium satellites. Source of nonparametric estimation: [15]

The 2-Weibull mixture function (equation (39)) deviates less than 0.5% from the original fit by Dubos et al. [15] over the whole observation window. As depicted in Figure 4-51 and Figure 4-52, the new parametric function follows the nonparametric data better up to one year in time. After that, the biggest deviations are located at five years, and at the end of the observation window, when the new fit uses the cut-off data while the original fit by Dubos et al. (equation (38)) uses the full data set. Overall, it can be noticed that infant mortality influences medium satellites to a lesser degree than small satellites. Further discussion of the data will follow in Chapter 5.

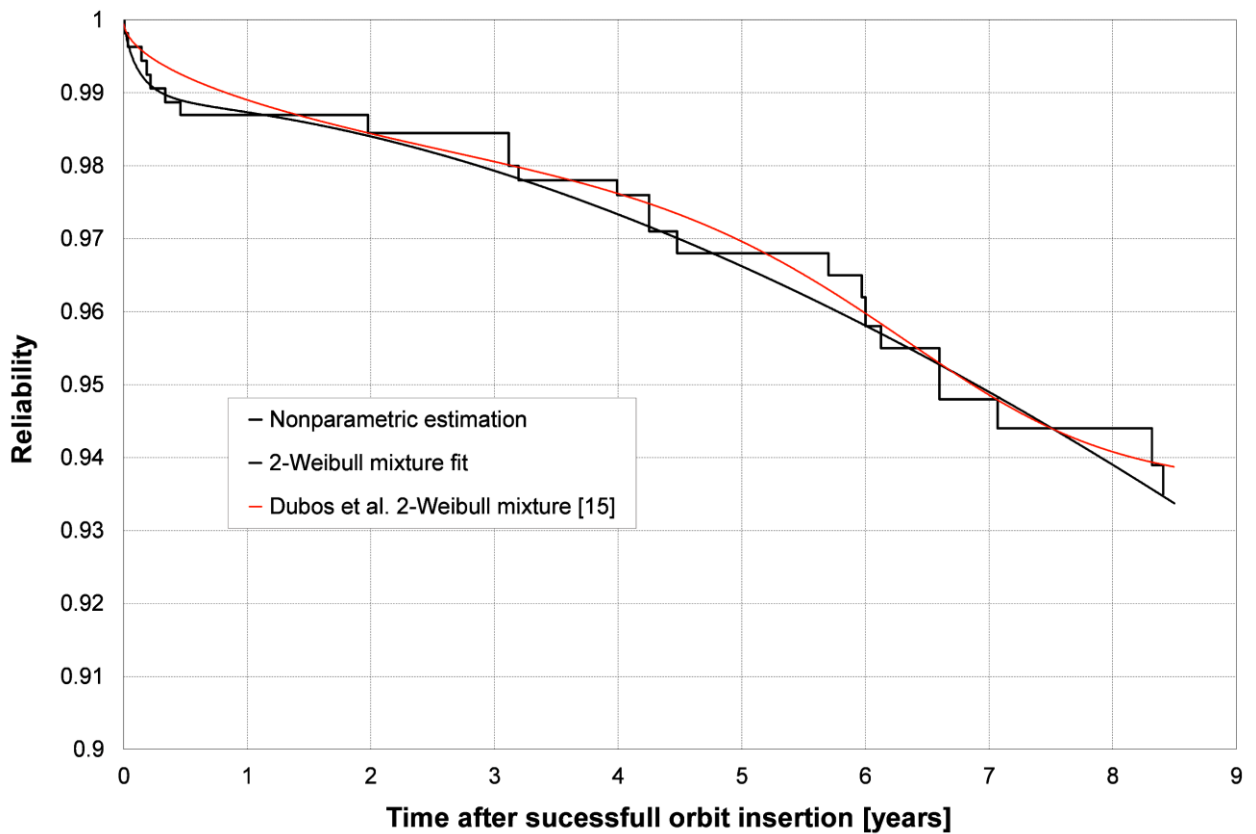


Figure 4-51: Comparison between 2-Weibull mixture fit (equation (39)) of the reduced nonparametric reliability data of medium satellites and the 2-Weibull mixture fit by Dubos et al. [15] (equation (38)). Source of nonparametric estimation: [15].

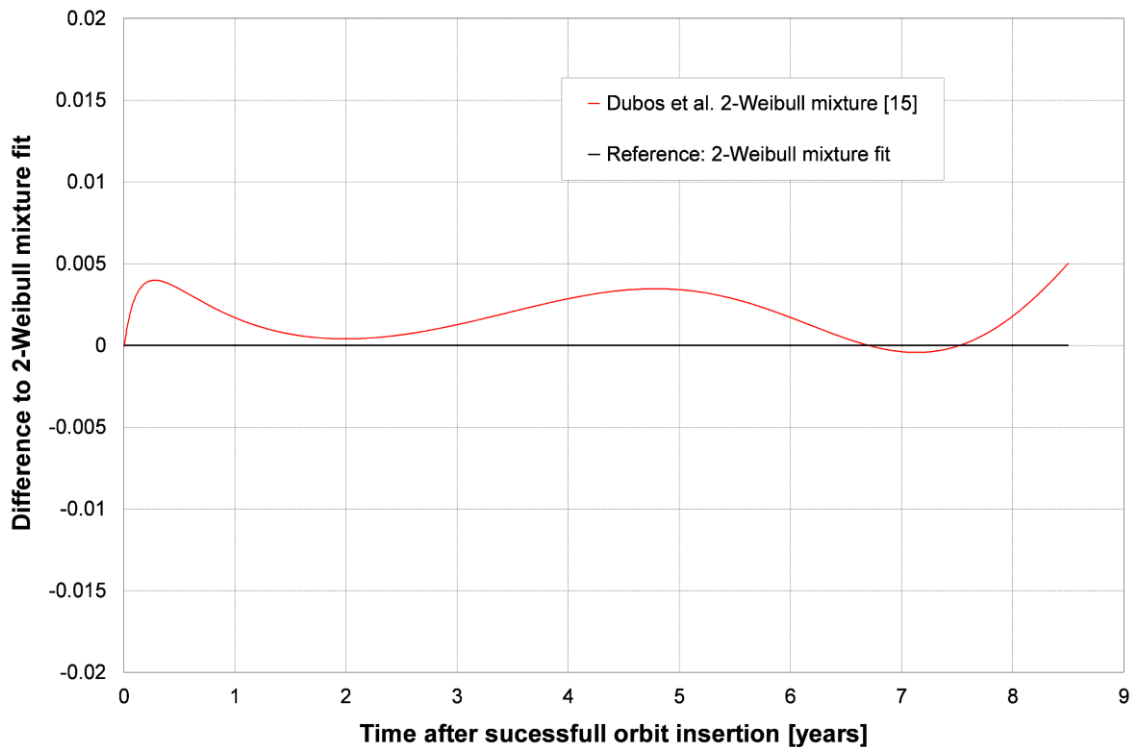


Figure 4-52: Difference of the 2-Weibull mixture model of Dubos et al. [15] (equation (38)) to new 2-Weibull mixture fit (equation (39)).

4.1.3 Analysis of Large Satellite Reliability

Finally, the group of large satellites was also studied by Dubos et al. [15]. As before, their 2-Weibull mixture function follows the nonparametric estimation very good, but consists of values not plausible for the different terms:

$$R(t) = 0.905 \cdot \exp \left[- \left(\frac{t [y]}{24,700} \right)^{0.3558} \right] + 0.095 \cdot \exp \left[- \left(\frac{t [y]}{11.9} \right)^{3.579} \right] \quad (40)$$

Instead of having a low-to-moderate scale factor for the infant mortality portion (shape factor $\beta_1 = 0.3558$) of the function, the scale factor of this term is $\theta_1 = 24,700$ years. As noted before, statistically that would mean that at $t = 24,700$ years, 63.2% of the group of satellites have failed due to this term, and still 36.8% will fail after that point in time. Extrapolation of the data up to this time point is of course not valid, but this also influences the fraction of satellites failed due to the infant mortality term within the observation window of 14.5 years, as shown in Figure 4-53 and Figure 4-54. The wear-out portion in this function has a more realistic appearance than for the small and medium category (Figure 4-55), but the number of satellites failed due to the infant mortality term continues to grow throughout the observation window.

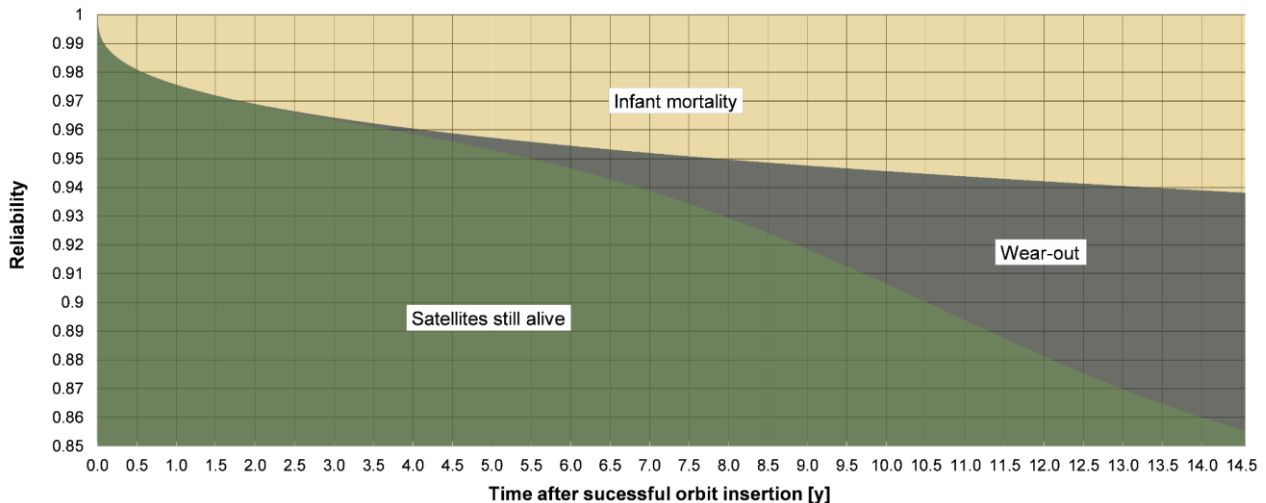


Figure 4-53: Fractions of large satellites that failed due to the two different terms of the 2-Weibull mixture function of Dubos et al. [15] (equation (40)). Green depicts satellites still alive, yellow satellites failed due to the infant mortality term ($\beta_1 = 0.3558$) of the function, and black satellites failed due to the wear-out term ($\beta_2 = 3.579$) of the function.

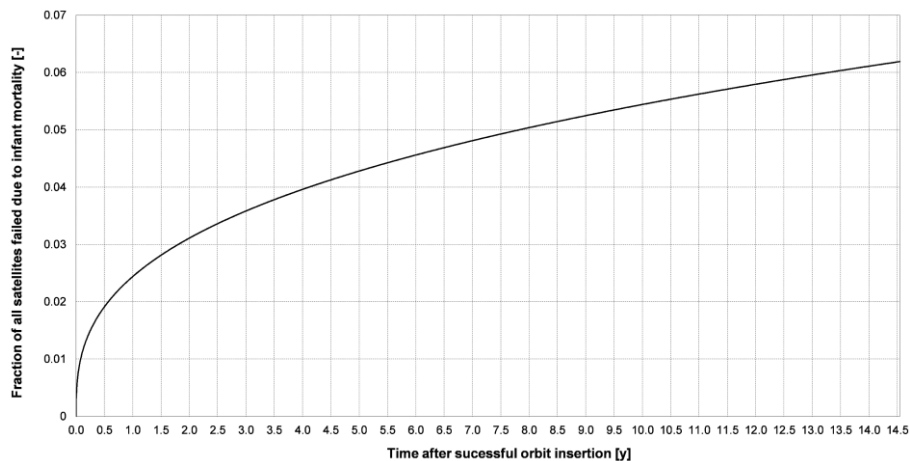


Figure 4-54: Fraction of all large satellites that failed due to the infant mortality term of the 2-Weibull mixture function by Dubos et al. [15] (equation (40)).

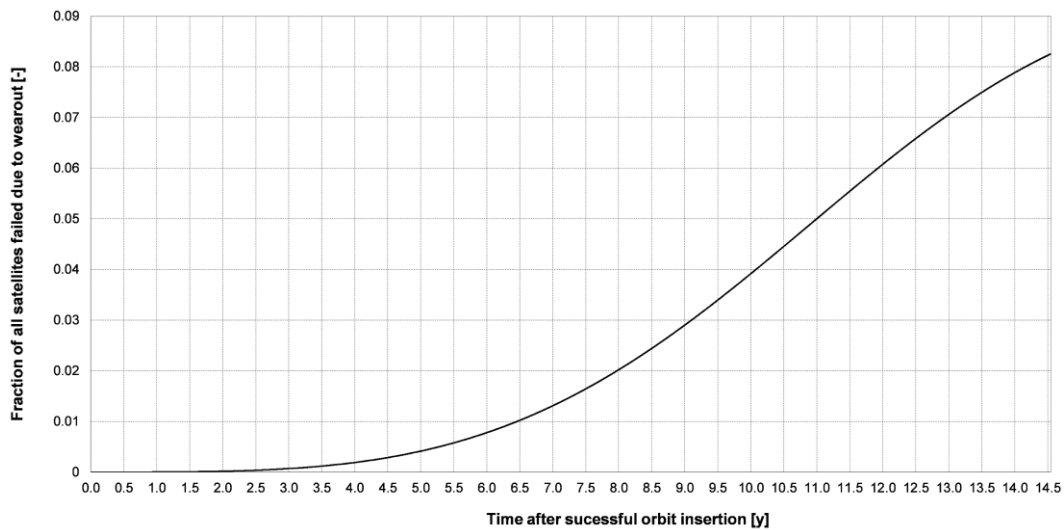


Figure 4-55: Fraction of all large satellites that failed due to the wear-out term of the 2-Weibull mixture function by Dubos et al. [15] (equation (40)).

Thus, to improve the parametric model, a new PNZ-modified 2-Weibull mixture fit was implemented on the nonparametric data. The PNZ modification stems from 2 large satellites of the studied group that failed within the first week. Since the activation and checkout of large satellites on-orbit usually needs a certain amount of time, dead on arrival/activation is a valid assumption for those two satellites. The nonparametric data of Dubos et al. [15] are almost identical for this class of satellites with the nonparametric data shown in Saleh & Castet [22], so the latter work was chosen for this model due to its availability in tables within their publication (so no graphical estimation was needed). The cutoff for the model was set to $t = 12$ years, and the shape factor of the first Weibull term was fixed to a value of one (constant failure rate). The parametric fit was estimated as:

$$R(t) = 0.9957 \cdot \left(0.0323 \cdot \exp \left[- \left(\frac{t}{1.077} \right)^1 \right] + 0.9677 \cdot \exp \left[- \left(\frac{t}{26.69} \right)^{2.598} \right] \right) \quad (41)$$

The function (see Figure 4-56) has a goodness-of-fit of $R^2 = 0.9937$ and the p_{NZ} value was set to 0.9957 (nonparametric reliability at $t = 7$ days after two satellites failed). The first term has a constant failure rate with a moderate scale factor θ_1 of 1.077 years. The second, more dominant term is a wear-out function with a shape factor of $\beta_2 = 2.598$ and a scale factor of $\theta_2 = 26.69$ years⁷⁴. As shown in Figure 4-57, this results in 0.43% ($1 - p_{NZ}$) of all satellites failing at activation, and 1.9% failing due to the constant failure rate term until $t = 1$ year. At this point in time, only a little fraction of reliability reduction is contributed by the wear-out term of the parametric model (0.018%). At $t = 7$ years, the constant failure rate term has grown to 3.2% and already saturated due to its moderate scale factor. This cannot be fully explained by a traditional constant failure behavior (i.e., random errors over the whole lifetime). The 95% confidence interval shows, that the constant failure rate term flattens out after $t = 7$ years in all scenarios (see Figure 4-58). Thus, infant mortality might be the underlying cause for this decline in reliability. At seven years, the wear-out function reaches 2.9% and continues to grow until the end of the observation window, when it reaches 11.3% (see Figure 4-59). In general, wear-out seems to be very apparent for this class of satellites, which is somehow expected since large satellites usually incorporate complicated attitude and energy systems and carry out more demanding missions than the other classes of satellites, and this degradation could result in the end of the mission.

⁷⁴ 95% confidence intervals: $\alpha_1 = (0.0244, 0.0403)$, $\alpha_2 = (0.9597, 0.9756)$, $\beta_2 = (2.139, 3.058)$, $\theta_1 = (0.5507 \text{ years}, 1.602 \text{ years})$, $\theta_2 = (23.04 \text{ years}, 30.33 \text{ years})$, β_1 was set constant to one.

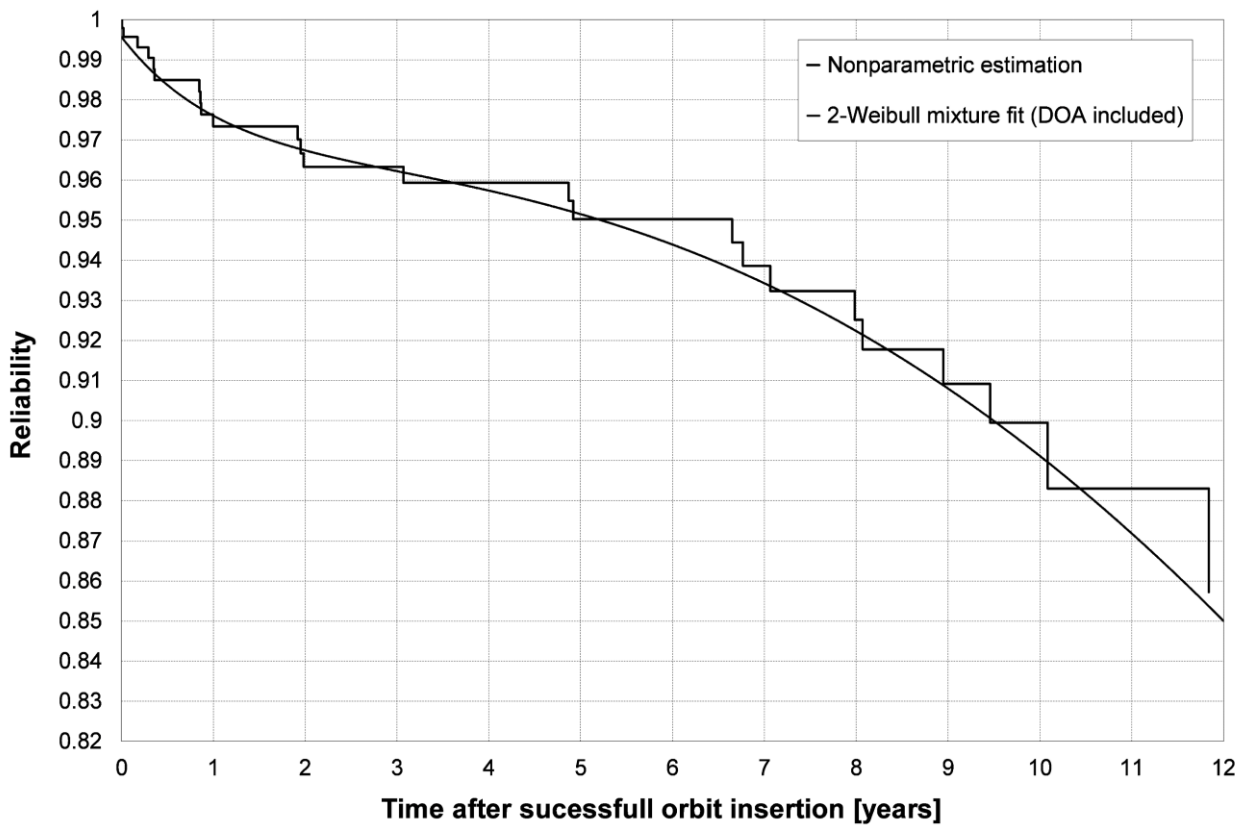


Figure 4-56: PNZ modified 2-Weibull mixture fit (equation (41)) of large satellite reliability. Data source of nonparametric estimation: [22].

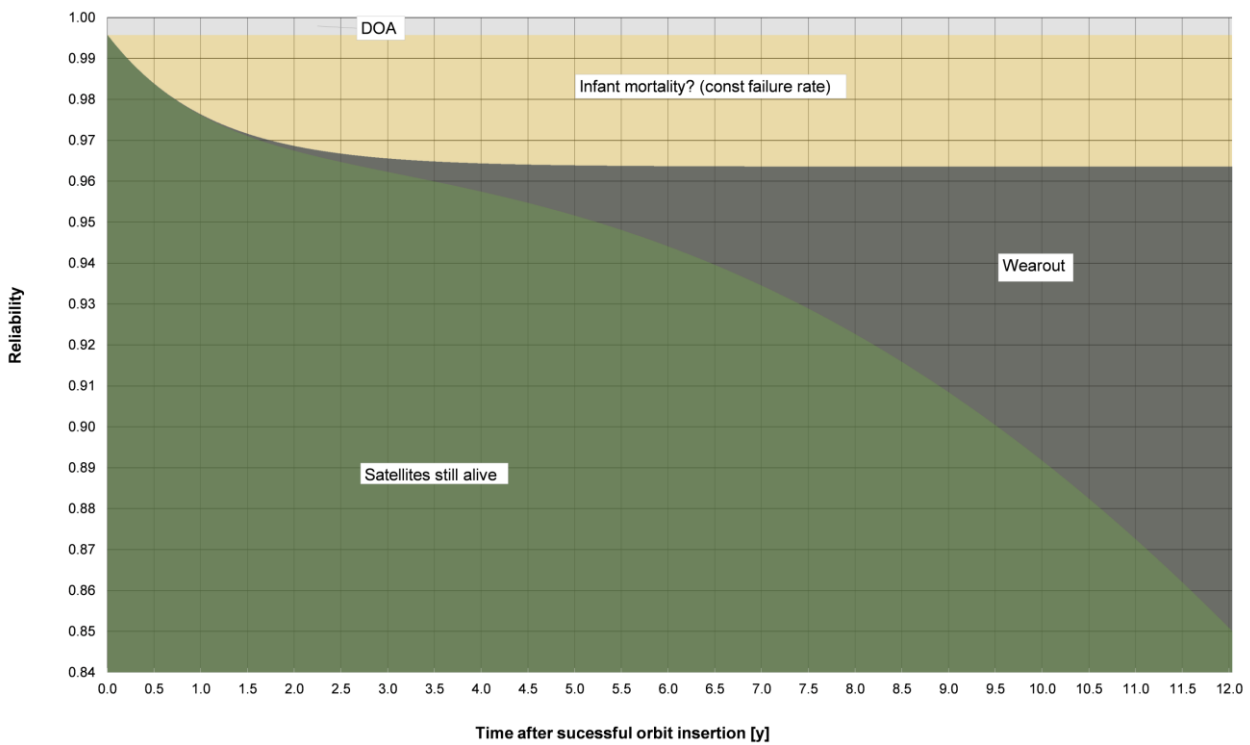


Figure 4-57: Fractions of large satellites failed due to the three different portions of the PNZ modified 2-Weibull mixture function (equation (41)) within the first 12 years on-orbit. Green depicts satellites still alive, grey satellites that failed on arrival/activation, yellow satellites that failed within the constant failure rate portion of the function (infant mortality is assumed), and black satellites failed due to the wear-out portion ($\beta_2 = 2.598$) of the function.

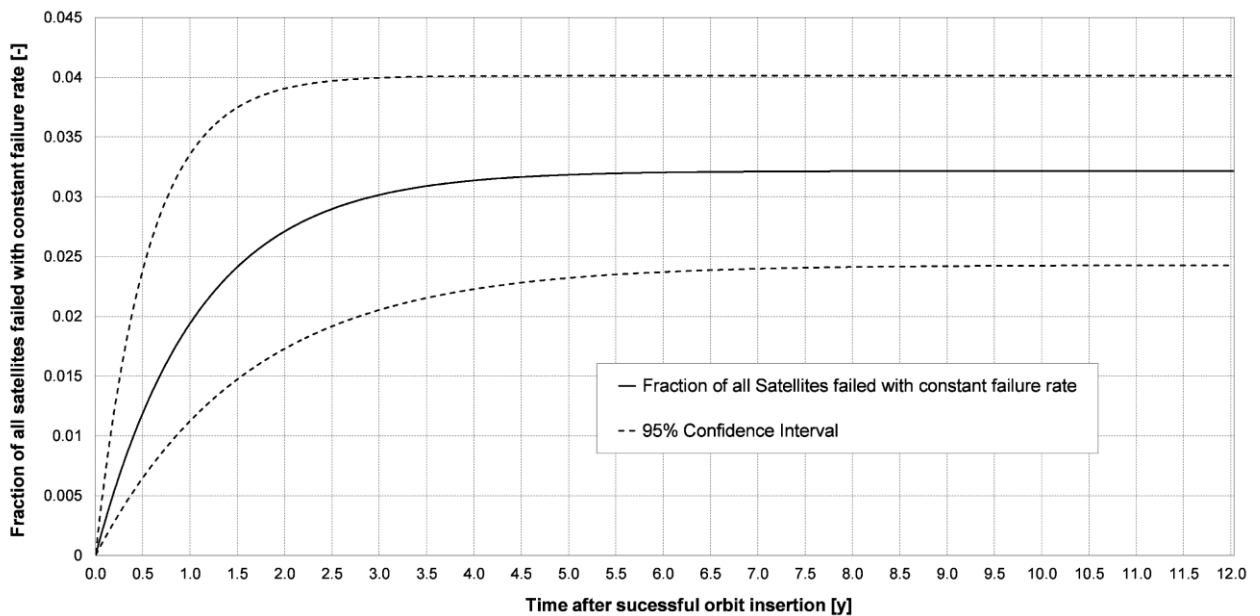


Figure 4-58: Fraction of all large satellites that failed due to the constant failure rate term of the PNZ-modified 2-Weibull mixture function (equation (41)).

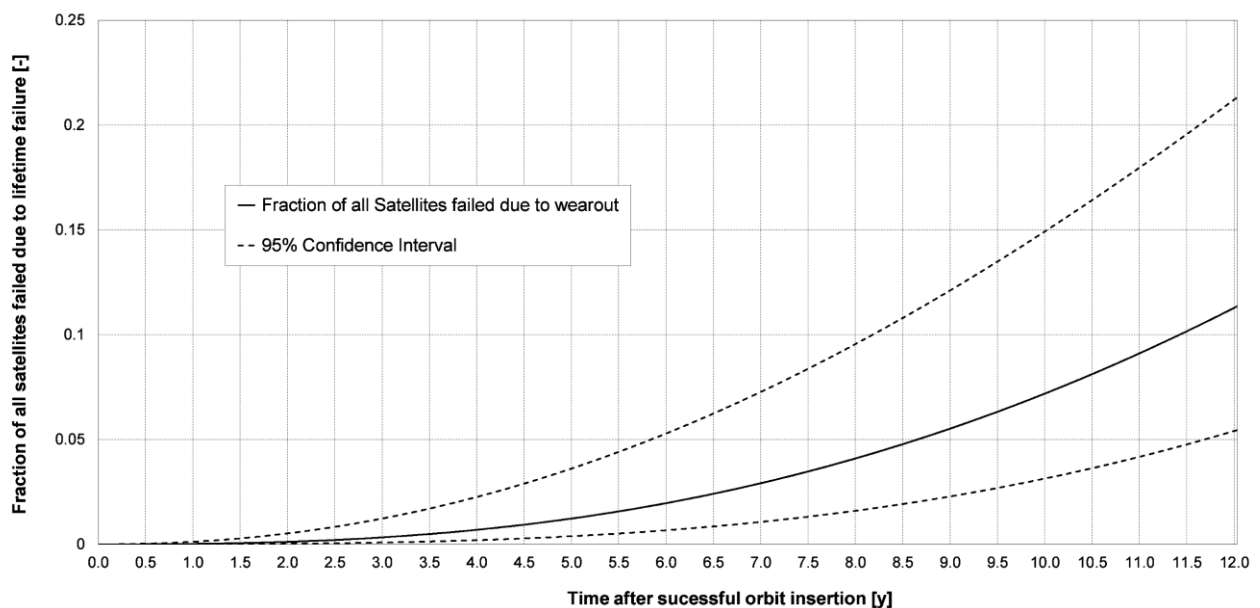


Figure 4-59: Fraction of all large satellites that failed due to the wear-out term of the PNZ-modified 2-Weibull mixture function (equation (41)).

The Weibull plot shows early deviations of the parametric fit to the nonparametric data, which can be attributed to the DOA-modification of the parametric function (Figure 4-60 left). The 25th percentile (-0.6%) and the 75th percentile (-0.01%) of the boxplot (Figure 4-60 right) and especially the whiskers show some dispersion, which can be explained mainly by the deviation to the nonparametric data between 10 and 12 years. This can also be seen in Figure 4-61. Generally, the new parametric fit follows the nonparametric estimation very good in the first five years, but slightly underestimates the reliability after that. The deviation to the fit by Dubos et al. starts growing after that point in time and reaches 3% at the end of the observation window. This can be partly explained by an additional failure point of the Dubos et al. nonparametric data at $t > 14$ years, which is not considered in equation (42). The model in equation (42) will be discussed in greater depth in Chapter 5.

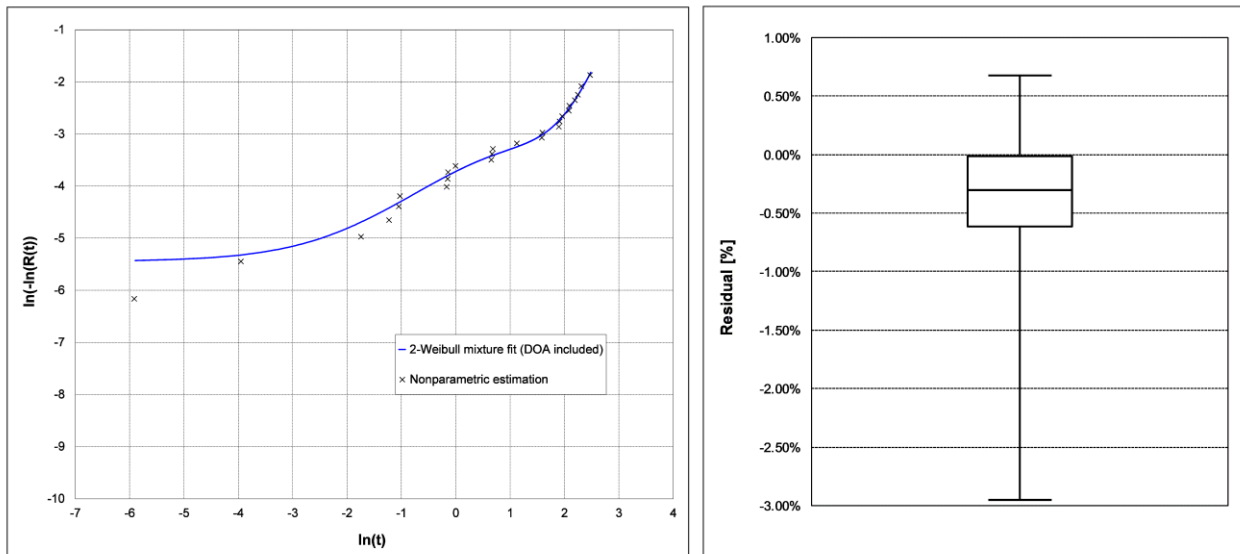


Figure 4-60: Weibull-plot (left) and boxplot (right) of the residuals between the PNZ-modified 2-Weibull mixture fit (equation (41)) and the reduced nonparametric estimation of large satellite reliability data. Source of nonparametric estimation: [22]

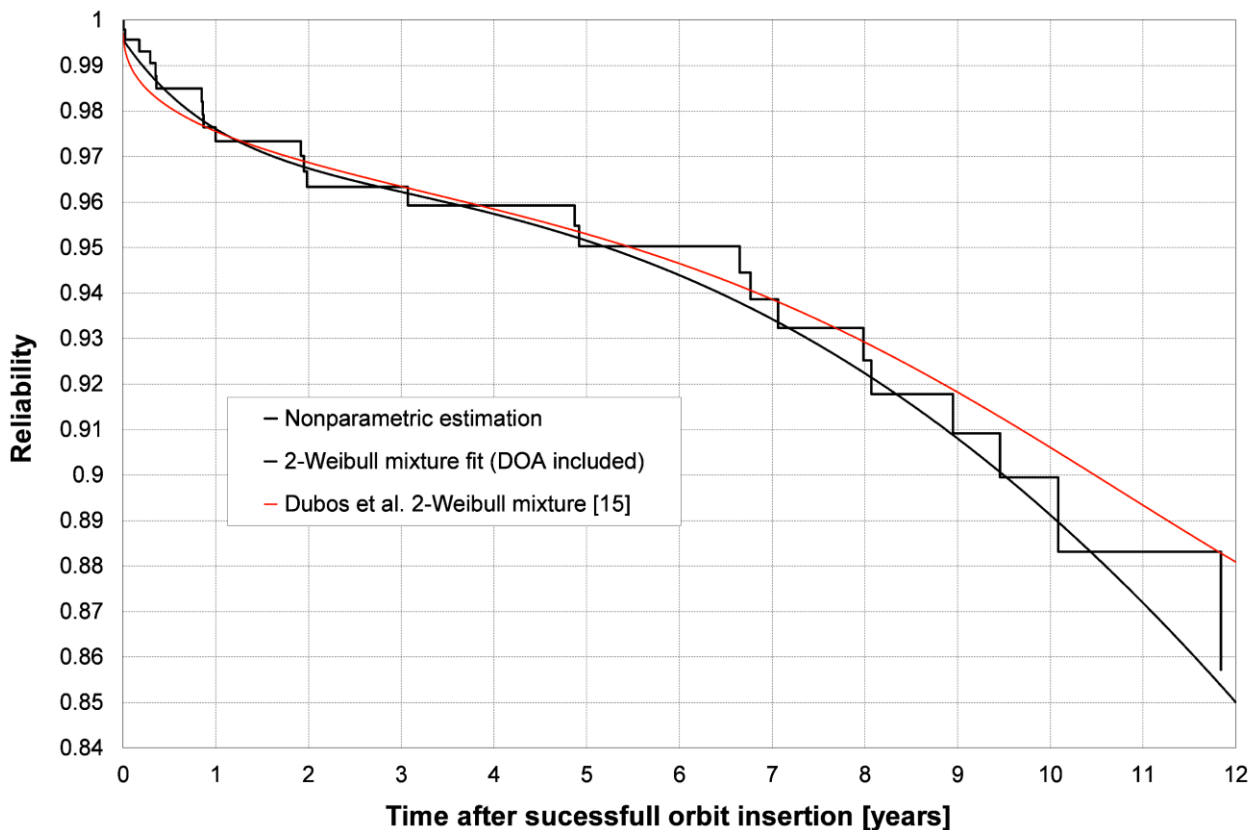


Figure 4-61: Comparison between PNZ-modified 2-Weibull mixture fit (equation (41)) and the 2-Weibull mixture fit by Dubos et al. [15] (equation (40)). Source of nonparametric estimation: [22].

This concludes the section on the analysis of general satellite reliability. As we have seen, the time-dependent failure behavior of satellites from today’s in-flight reliability data can be extracted, but the parameters of the Weibull models must be chosen carefully. The next section will deal with the time-dependent on-orbit failure behavior of CubeSats.

4.2 CubeSat Reliability

This section is an extended and adapted version of two conference papers ([12] and [264]) by the author of this thesis.

As the second goal of this thesis, CubeSat in-flight reliability data were collected, and the time-dependent failure behavior extracted. This section deals with the findings of this effort, which started in early 2014, and led to one publication in 2016 and one publication in 2017. The results of these papers and further, new findings are presented in the following. As we already have seen in Section 2.3, Swartwout showed the causes and the success and failure rates of past CubeSat missions in multiple publications [11], [247], [248], [249], [251], [250] and in an online database [5], yet the time dependence of both parameters remained unknown. To fill this gap, the CFDB (Table 4-1) was built in late 2014. It is comprised of 178 individual CubeSats up to a launch date of 06/30/2014 and was created with the aim to collect time of failure and root-cause data of failure for all CubeSats launched so far. For this purpose, information was collected from publicly available sources [265], [266], [267], [268], [269], [270] as well as from work by Klofas, Anderson & Leveque [271] and Klofas & Leveque [272] and numerous publications on the individual spacecraft. Furthermore, information was gathered within a survey, which was sent out in late 2014 to 987 individuals affiliated with CubeSat programs worldwide, and fully answered by 113. Finally, through personal communication during conferences or via E-Mail, yet unpublished information was also added to the database. The first version of the database was completed by the end of 2015, containing the class, the sub-type, the launch date, the time of failure and the root cause of 70 failures within 178 missions, not including launch failures. Furthermore, in the case of successful on-orbit arrival, the censored time of the CubeSats (i.e., the point in time they are retired or the observation window ends) can be accessed. Since the publicly available information on satellites of the Flock Constellation of Planet Labs was scarce, those satellites were not included in the database.

Table 4-1: The CFDB.

Satellite Name	Class	Sub-type	Launch	Time of Failure	Cause	Censored Time (no failure occurred)
CubeSat 1	uni	1U	30.06.2003	22.09.2003	XYZ	-
CubeSat 2	uni	2U	30.06.2003	-	-	30.09.2013
...
CubeSat 178	uni	3U	30.06.2014	-	-	31.12.2014

Table 4-2 summarizes the failure times of all CubeSats included in the database. As already pointed out, all CubeSats launched up to 06/30/2014 are included in the database and the observation window ended on 12/31/2014. Due to scarcity of data for longer time periods, the observation window is mostly limited to 12 months in this work, and only occasionally extended to 1.6 years⁷⁵.

Table 4-2: Failure times (in days) of all CubeSats launched up to 06/30/2014 within an observation window of 1.6 years (i.e., if the failure time was beyond 584 days, it is not mentioned in here).

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	5
5	7	8	10	10	10	23	28	30	30	34	34
39	40	46	60	60	61	80	84	92	97	115	124
124	138	203	207	314	334	363	569				

⁷⁵ Since there is only one documented failure in the group of 178 CubeSats in between one year and 1.6 years, this is mainly for sensitivity analysis purposes. The general observation timeframe could be further extended in future work, if more CubeSats outlive their first year on-orbit and thus more data become available.

Data from the CFDB were subsequently used for nonparametric and parametric reliability analysis. As shown for example in the work of Saleh & Castet [22] the Kaplan-Meier estimator [273] is best suited for nonparametric reliability analysis and samples with the type of censoring occurring in the CFDB. The Kaplan-Meier estimator for reliability $R(t)$ for censored data used in this study is adapted from [116] :

$$R(t) = \prod_{i=1}^k \frac{n_i(t_i) - 1}{n_i(t_i)}; \quad t(i) \leq t \quad (42)$$

with t_i as the time to i^{th} failure, n_i as the number of operational units right before t_i , and k the number of failures up to t . More details on the background of nonparametric analysis for satellite reliability data can be found in [22]. Figure 4-62 shows the results of the nonparametric reliability estimation with 95% confidence intervals for one year on-orbit. As it can be seen, the overall reliability of CubeSats is strongly dominated by DOA cases, in which the satellite was ejected from its deployer and subsequently never achieved a detectable functional state. Due to these DOA cases after a successful deployment, the overall reliability drops instantly to a value of 81.5% (95% confidence interval between 87% and 75.6%). After 100 days on-orbit, a reliability value of 66.3% (73% and 59% as the 95% confidence interval) shows that infant mortality is the dominant effect for CubeSats. At the end of the observation window of one year, the nonparametric reliability declines to 59.4%. Thus, the reliability of the studied group of CubeSats decreases only by 6.9% between day 100 and day 365. Besides DOA and infant mortality, this means that CubeSats in LEO are not yet as susceptible to wear-out as for example geostationary satellites, but this effect might emerge in the future with longer lifetimes of future CubeSats. Also, the data shown in this thesis is cut off at a point in time that might be too early to see wear-out effects. Thus, statistical significant data over a longer period of time could help to substantiate or disprove wear-out of CubeSats in the future.

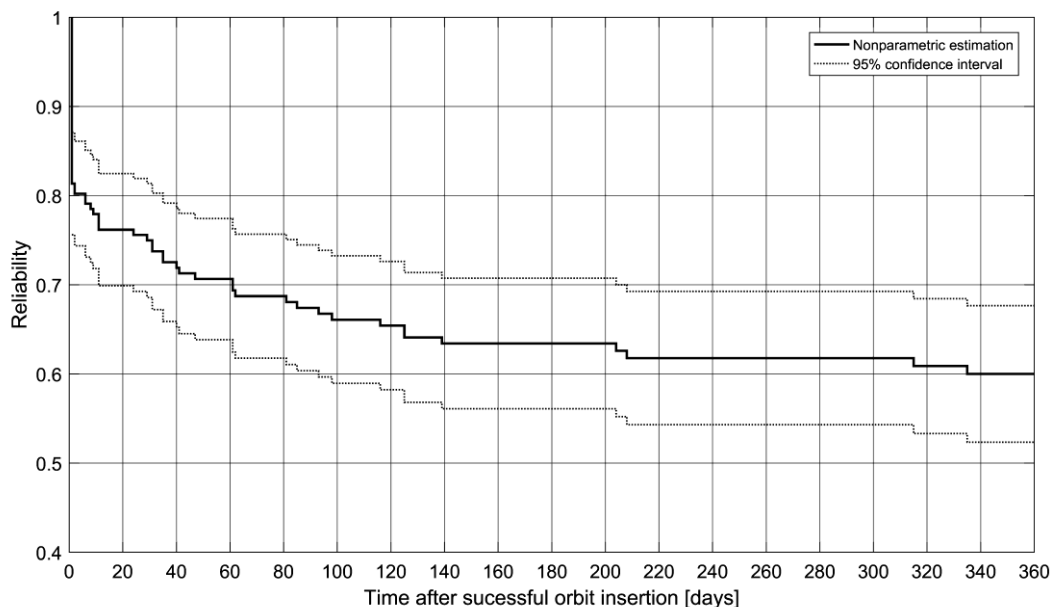


Figure 4-62: Nonparametric estimation of CubeSat reliability and 95% confidence interval within an observation window of one year.

Since parametric models can be used in a broader range of applications, it was decided to again create a parametric function resembling the nonparametric reliability estimation. As before, the Weibull distribution was chosen for this purpose. To determine the parameters of the Weibull function, the nonlinear least-squares method and later the MLE method are used. Looking at the nonparametric reliability of CubeSats, with their large fraction of DOAs, the p_{NZ} modification factor must be applied to any parametric fit. As a first step, a Single-Weibull function was estimated with the following parameter:

$$R(t) = 0.8146 \cdot \exp \left[- \left(\frac{t[y]}{13.08} \right)^{0.4321} \right] \quad (43)$$

The goodness-of-fit of this function is $R^2 = 0.9651$, and though it is only a Single-Weibull fit, it follows the nonparametric estimation quite well, as can be seen in Figure 4-63. The p_{NZ} modification factor resembles the high DOA-rate of 18.54%. The Weibull term has a shape factor resembling a decreasing failure rate of $\beta = 0.4321$. The scale factor θ is 13.08 years, which is a high value considering the average lifetime of CubeSats. As before, no extrapolations beyond the 18 months of maximum observation window should be made, as the statistical data used for this study is scarce beyond this point in time. The modelled parametric function shows the two main prevalent problems with CubeSats: DOA, thus satellites that never achieved a functional state, and high infant mortality, i.e., satellites that fail early due to the reasons mentioned in earlier chapters. As depicted in Figure 4-64, the reliability of CubeSats drops by more than 22% within their first year on-orbit, and that is on top of the already high rate of DOAs.

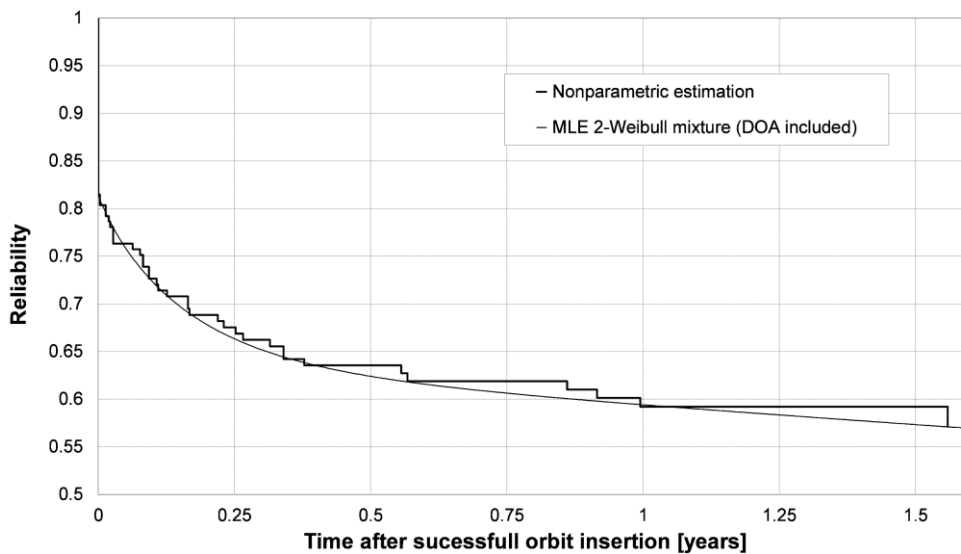


Figure 4-63: Nonparametric reliability and PNZ-modified Single-Weibull fit (equation (43)) of CubeSat reliability within their first 1.6 years on-orbit.

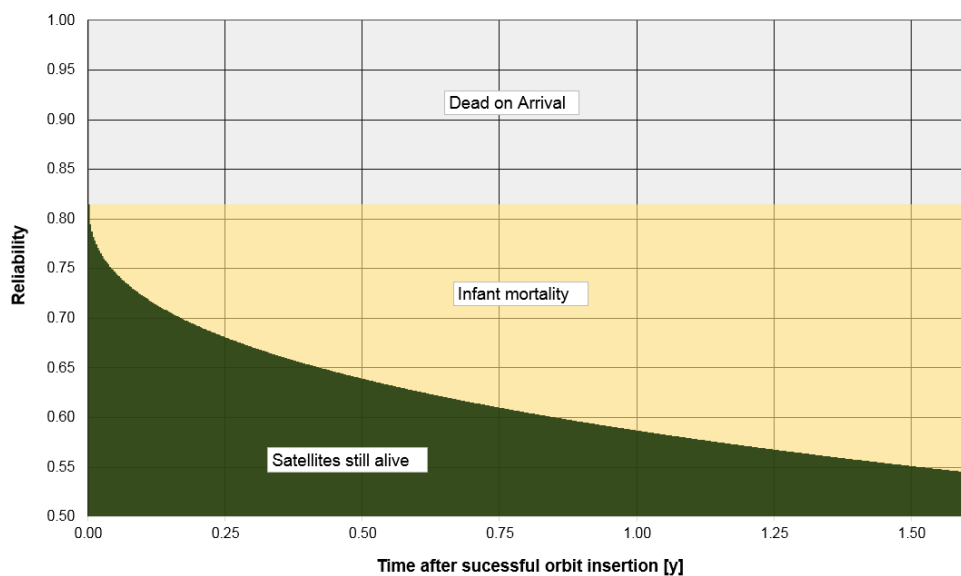


Figure 4-64: Fractions of CubeSats failed due to the two different portions of the PNZ-modified Single-Weibull function (equation (43)) within their first 1.6 years on-orbit. Green depicts satellites still alive, grey satellites that failed on arrival/activation and yellow satellites that failed due to infant mortality ($\beta = 0.4321$).

The 95% confidence interval⁷⁶ (Figure 4-65) shows that CubeSats experience infant mortality, and the infant mortality term is not flattening out within the observation window. Thus, there is still room for improvement of the parametric fit, and this can also be seen in the Weibull-Plot (Figure 4-66 left) and the boxplot, in which the 25th percentile is at -2.3% and the 75th percentile at 0.28% (Figure 4-66 right).

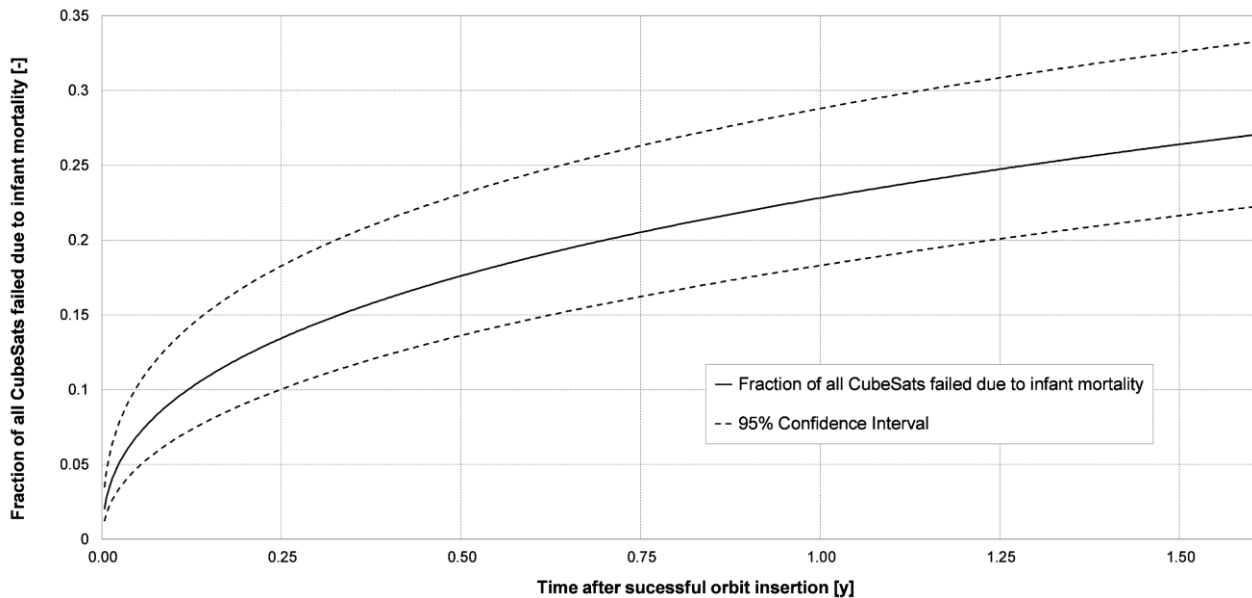


Figure 4-65: Fraction of all CubeSats that failed due to the infant mortality term of the PNZ-modified Single-Weibull function (equation (43)).

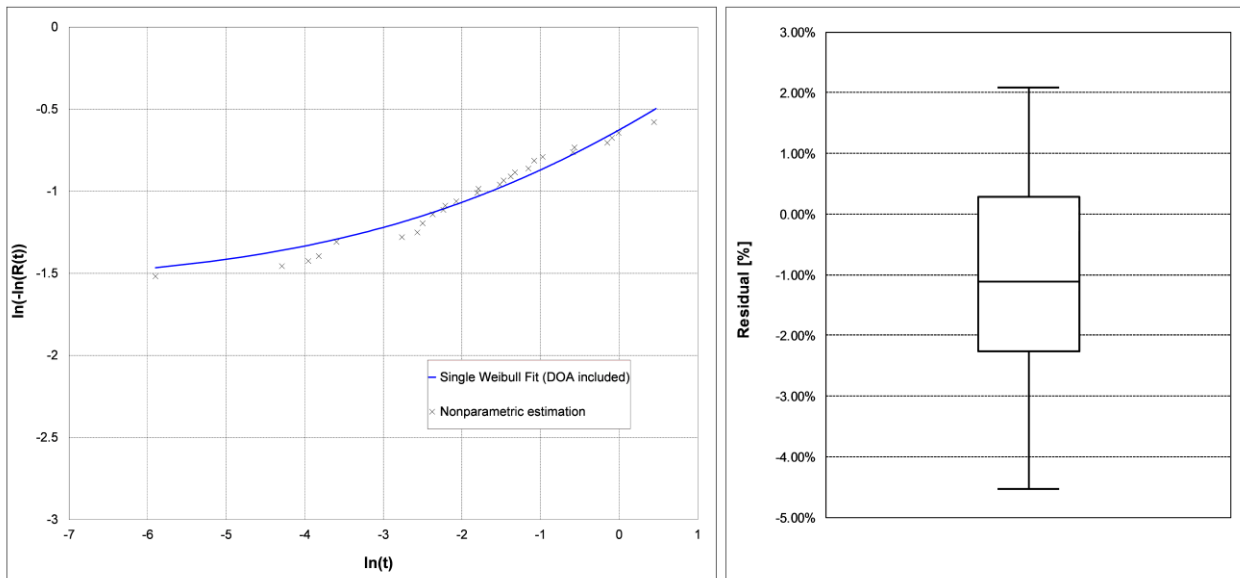


Figure 4-66: Weibull-plot (left) and boxplot (right) of the residuals between the PNZ-modified Single-Weibull fit (equation (43)) and the nonparametric estimation of CubeSat reliability.

Thus, as a next step, a 2-Weibull mixture function was applied to the nonparametric data. Using the nonlinear-least squares approach the PNZ-modification was again incorporated in the function due to the high rate of DOAs. Besides the infant mortality term, a constant failure rate term was introduced in the 2-

⁷⁶ 95% confidence interval β : 0.3891, 0.475; 95% confidence interval θ : 8.399 years, 17.76 years.

Weibull mixture function, due to the earlier assumption that the studied group of CubeSats experiences no wear-out in the observation window of maximum 1.6 years. The resulting function is:

$$R(t) = 0.8146 \cdot \left(0.2248 \cdot \exp \left[- \left(\frac{t [y]}{0.1705} \right)^{0.8327} \right] + 0.7752 \cdot \exp \left[- \left(\frac{t [y]}{15.86} \right)^1 \right] \right) \quad (44)$$

Besides the second Weibull function, the reduced scale factor and slightly larger shape factor of the infant mortality portion of the function are the most obvious changes to the Single-Weibull fit shown before. With $\beta_1 = 0.8327$ (95% confidence interval of 0.7211 and 0.9444) and $\theta_1 = 0.1705$ years (95% confidence interval of 0.1146 years and 0.2264 years) the function still implements a decreasing failure rate and describes satellites that failed early in life. The second term describes a constant failure rate function with a scale factor of $\theta_2 = 15.86$ years (95% confidence interval of 5.095 years and 26.63 years)⁷⁷. Overall, the PNZ-modified 2-Weibull mixture function (see Figure 4-67) has a goodness-of-fit of $R^2 = 0.9934$.

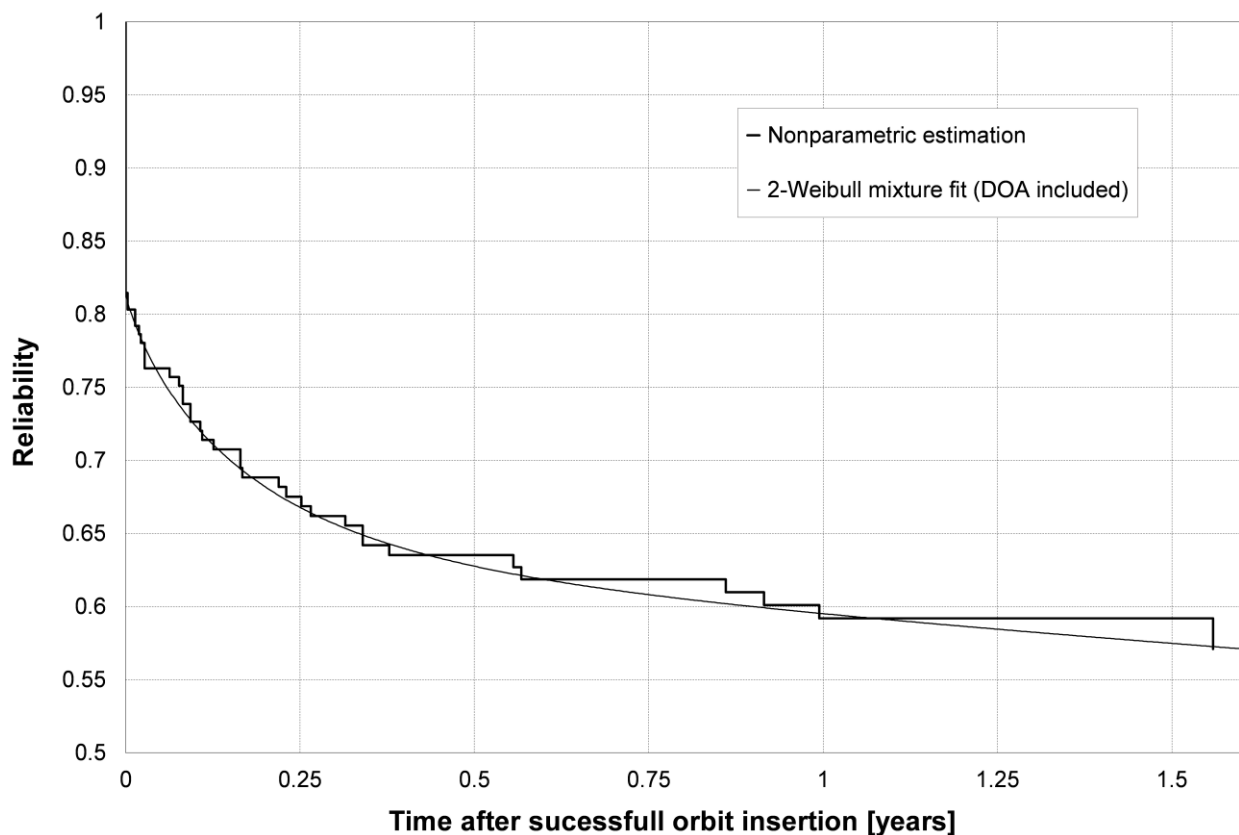


Figure 4-67: Nonparametric reliability and PNZ-modified 2-Weibull mixture fit (equation (44)) of CubeSat reliability within their first 1.6 years on-orbit.

The reduced scale factor leads to a more realistic time for saturation of the infant mortality rate, as can be seen in Figure 4-68 and Figure 4-69. Furthermore, Figure 4-68 shows that at $t = 1$ year, both DOA and infant mortality reduce the reliability of CubeSats by approximately 18% each (DOA = 18.5%, infant mortality = 18.1%). At that point in time, failures with constant failure rate account for approximately 3.8% reliability reduction. At the end of the observation window, the values change to 18.3% for infant mortality and 6% for the constant failure rate. Thus, the group of satellites that survived one year on-orbit, does not experience infant mortality anymore, as it also can be seen in Figure 4-69. This is in accordance with both,

⁷⁷ 95% confidence interval of α_1 is 0.1836 and 0.266. Thus, the 95% confidence interval of α_2 is 0.734 and 0.8164.

data from CubeSats after one year, but also with historical missions, as seen in Subsection 2.1.3. If satellites survived the early time in-orbit, there was a high chance that the missions fulfilled and sometimes surpassed their design lifetime. As shown earlier, this is due to most of the early failures being caused by engineering flaws, design errors and systematic faults, and this is an important lesson learned from the CubeSat reliability data shown in here. Failure with random error rates are only a little fraction of the failed CubeSats. Mainly, on-orbit failures of CubeSats are DOA-cases and infant mortality. We will deal with that in more detail at the end of this section and in Section 4.3.

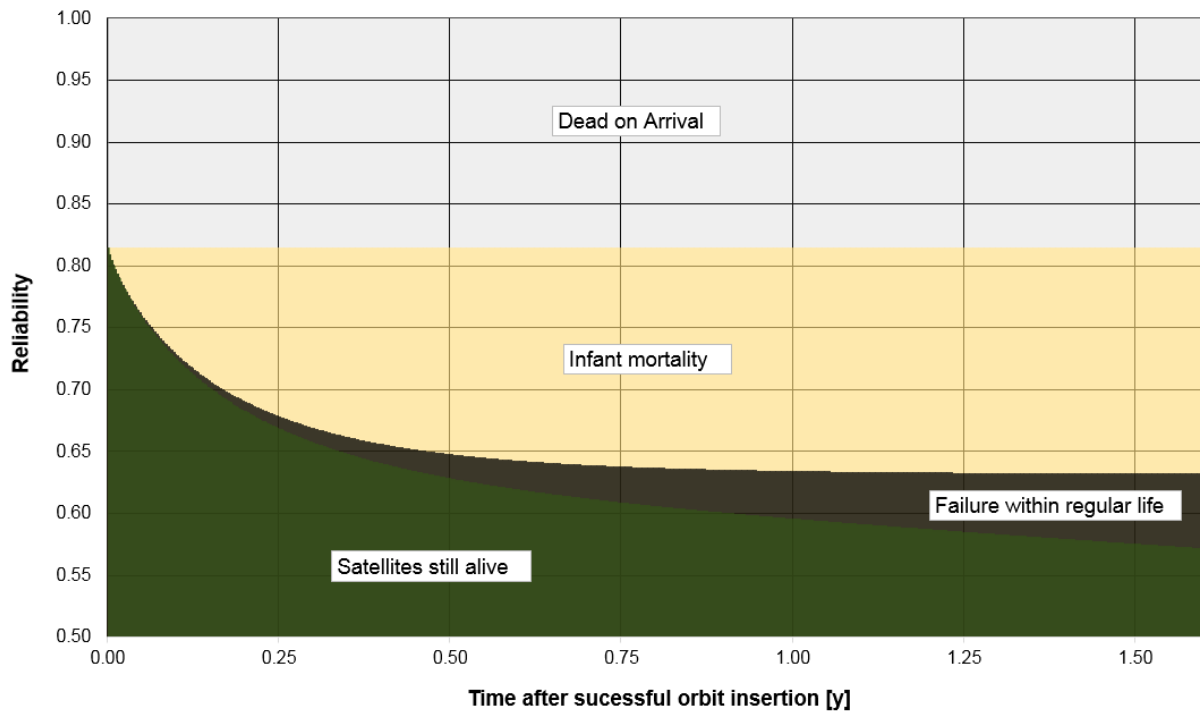


Figure 4-68: Fractions of CubeSats failed due to the three different portions of the PNZ-modified 2-Weibull mixture function (equation (44)) within their first 1.6 years on-orbit. Green depicts satellites still alive, grey satellites that failed on arrival/activation, yellow satellites that failed due to infant mortality ($\beta = 0.4321$) and black satellites that failed due to the constant failure rate portion of the function (i.e., random errors).

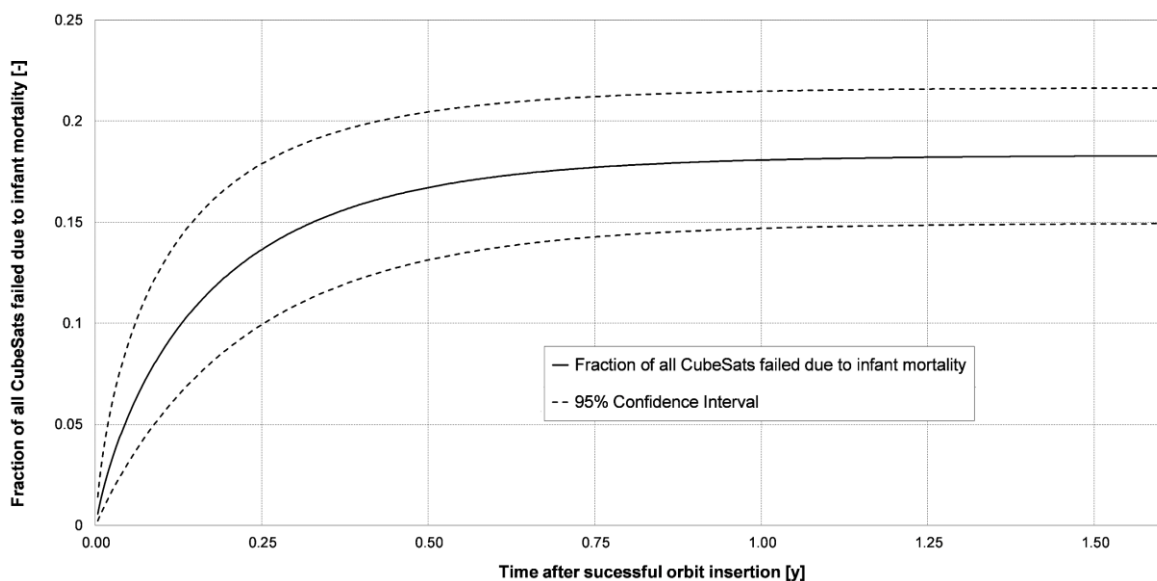


Figure 4-69: Fraction of all CubeSats that failed due to the infant mortality term of the PNZ-modified 2-Weibull mixture function (equation (44)).

The Weibull-Plot (Figure 4-70 left) and the boxplot (Figure 4-70 right) show a better fit to the nonparametric data than the Single-Weibull function. The 25th percentile is at -1.06% and the 75th percentile at -0.14%.

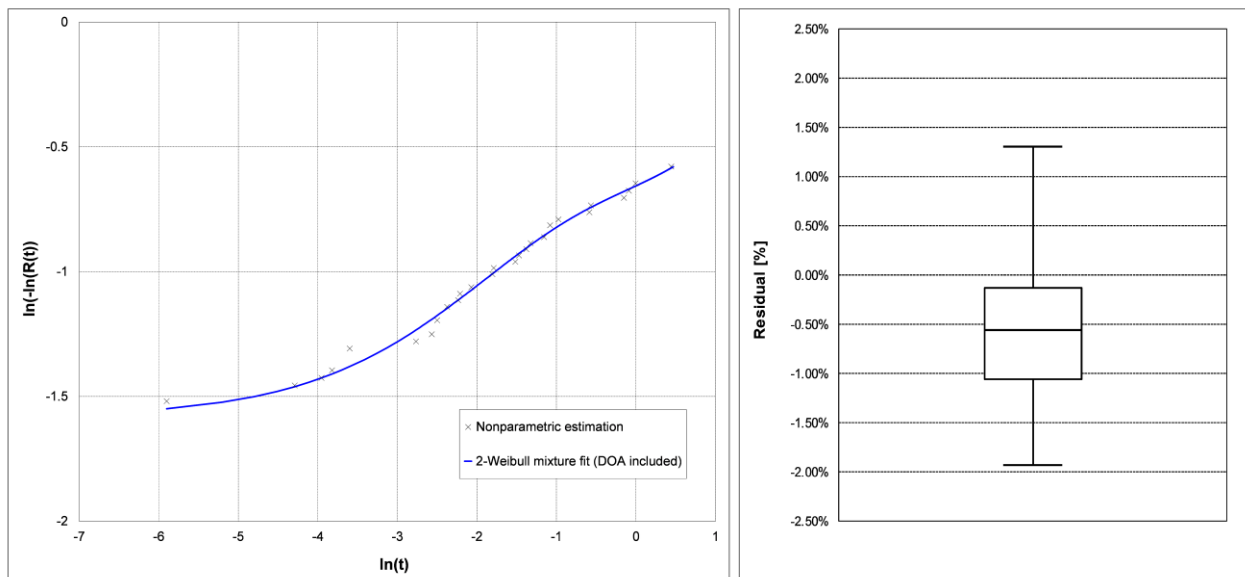


Figure 4-70: Weibull-plot (left) and boxplot (right) of the residuals between the PNZ-modified 2-Weibull mixture fit (equation (44)) and the nonparametric estimation of CubeSat reliability.

Thus, the 2-Weibull mixture fit shows sufficient accuracy on the nonparametric data, and no fit with more parameters is needed. As a next step, to study the sensitivity of the fit, the observation window was reduced to one year. For the one year of nonparametric data, a similar 2-Weibull mixture fit was estimated with the nonlinear least-squares approach:

$$R(t) = 0.8146 \cdot \left(0.262 \cdot \exp \left[- \left(\frac{t \text{ [y]}}{0.2158} \right)^{0.7975} \right] + 0.738 \cdot \exp \left[- \left(\frac{t \text{ [y]}}{57.91} \right)^1 \right] \right) \quad (45)$$

Due to the reduced data, the 95% confidence intervals of all parameters expand, and the scale factors θ_1 and θ_2 grow, but the overall characteristics of parametric fit remain, as can be seen in Figure 4-71.

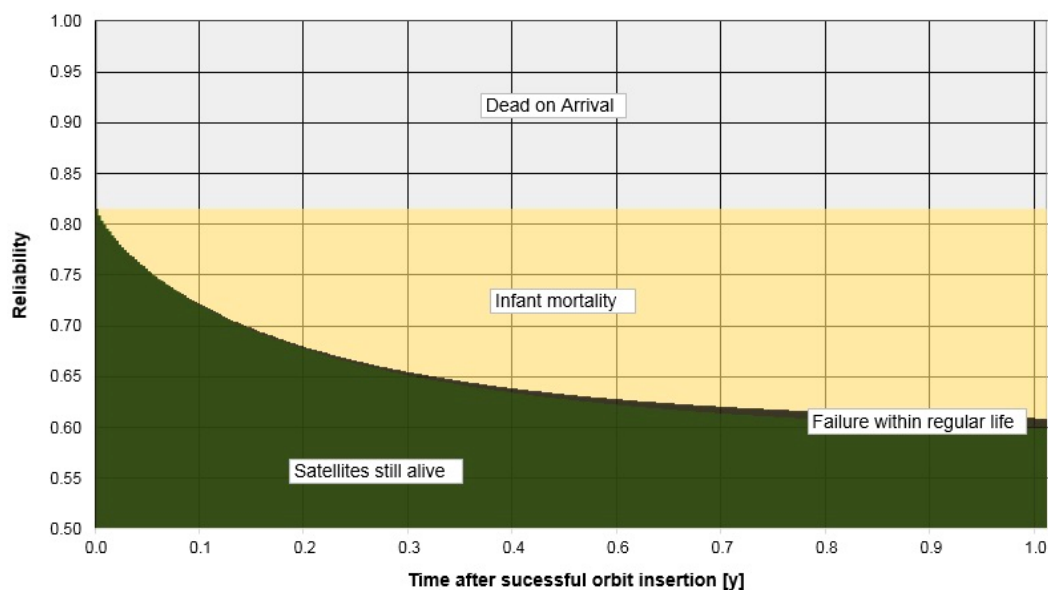


Figure 4-71: Fractions of CubeSats failed due to the three different portions of the PNZ-modified 2-Weibull mixture function (equation (45)) within the reduced observation window of one year.

Similar to the full fit, the fit of the reduced observation window shows the dominance of DOA and infant mortality for CubeSats, with an additional little fraction of constant failure rate cases. The goodness-of-fit of the functions stays at a relatively high value of $R^2 = 0.9934$. At $t = 1$ year, the same rate of DOA as before (same p_{NZ} modification factor) and slightly more infant mortality cases (20.6%) are modelled by this function. Thus, as the overall reliability is nearly the same for both fits (full fit: 40.5%, new fit: 40.2%, see Figure 4-72), the constant failure rate accounts only for 1% reliability reduction. Overall, this variation is not significant since the pattern of dominant DOA and infant mortality is not altered by it.

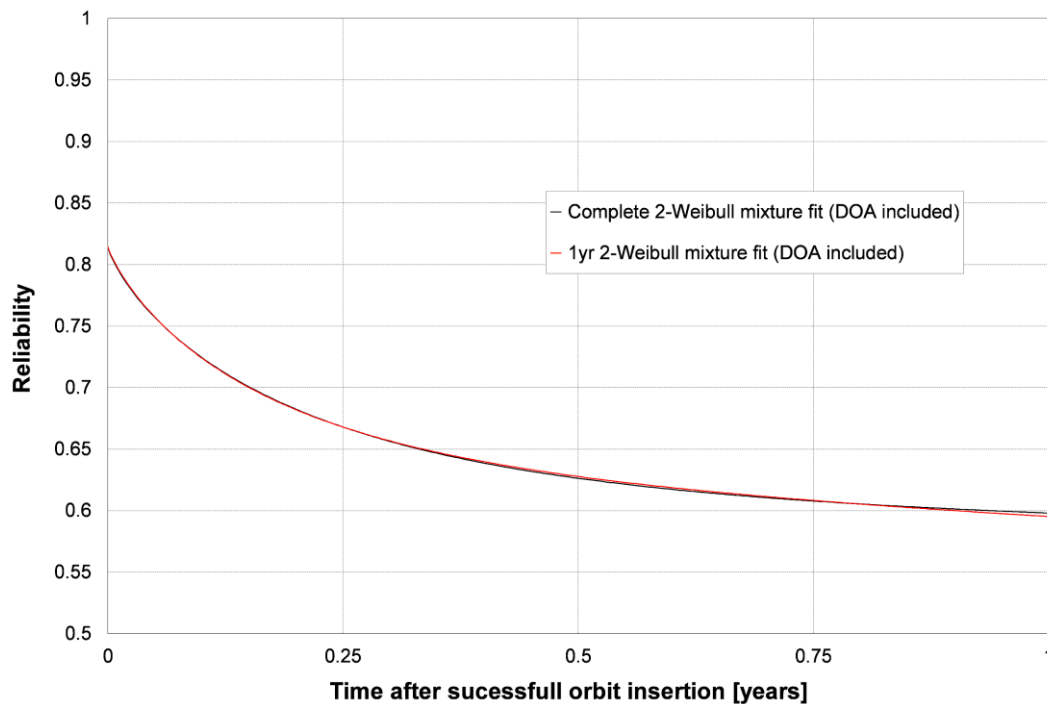


Figure 4-72: PNZ-modified 2-Weibull mixture fit using the complete observation window of 1.6 years (black) (equation (44)) and the reduced observation window of one year (orange) (equation (45)). The variation is below 0.5% within the observation window of one year.

Also, the overall results by the nonlinear least-squares method were verified by using the MLE method on the full set of data (cutoff at 1.6 years). The MLE function is:

$$R(t) = 0.8146 \cdot \left(0.2115 \cdot \exp \left[- \left(\frac{t [y]}{0.1587} \right)^{0.9017} \right] + 0.7885 \cdot \exp \left[- \left(\frac{t [y]}{13.244} \right)^{1.071} \right] \right) \quad (46)$$

As before, the failure rate is dominated by DOAs and infant mortality cases, which are resembled by the PNZ-modification and the first Weibull function ($\beta_1 = 0.9017$). For the MLE, the shape factor of the second Weibull function was also left variable. A shape factor of $\beta_2 = 1.071$ indicates that the before assumed constant failure rate was the right approach. The MLE parametric fit never deviates more than 1% from the nonlinear least squares estimated fit, as showed in Figure 4-73. Overall, the MLE-fitted function shows less deviation from the nonparametric data, with a 25th percentile of the residuals at -0.36% and 75th percentile at 0.56%. As presented in Figure 4-74, the rate of CubeSats failed due to infant mortality reaches 17.1% at $t = 1$ year and thus is 1% below the rate estimated by the nonlinear least-squares function. At the end of the observation window, this portion remains at the same value, while 6.3% of all satellites failed due to the (near) constant failure Weibull function. Overall, the MLE method estimates a reliability of 57.9% at $t = 1.6$ years, which is 0.8% more than estimated by the nonlinear least-squares approach. This deviation is small enough to prove that both approaches are valid for our purposes. Overall the high rate of DOA and infant

mortality is substantiated by the data, and this influences the methods to enhance the reliability of CubeSats.

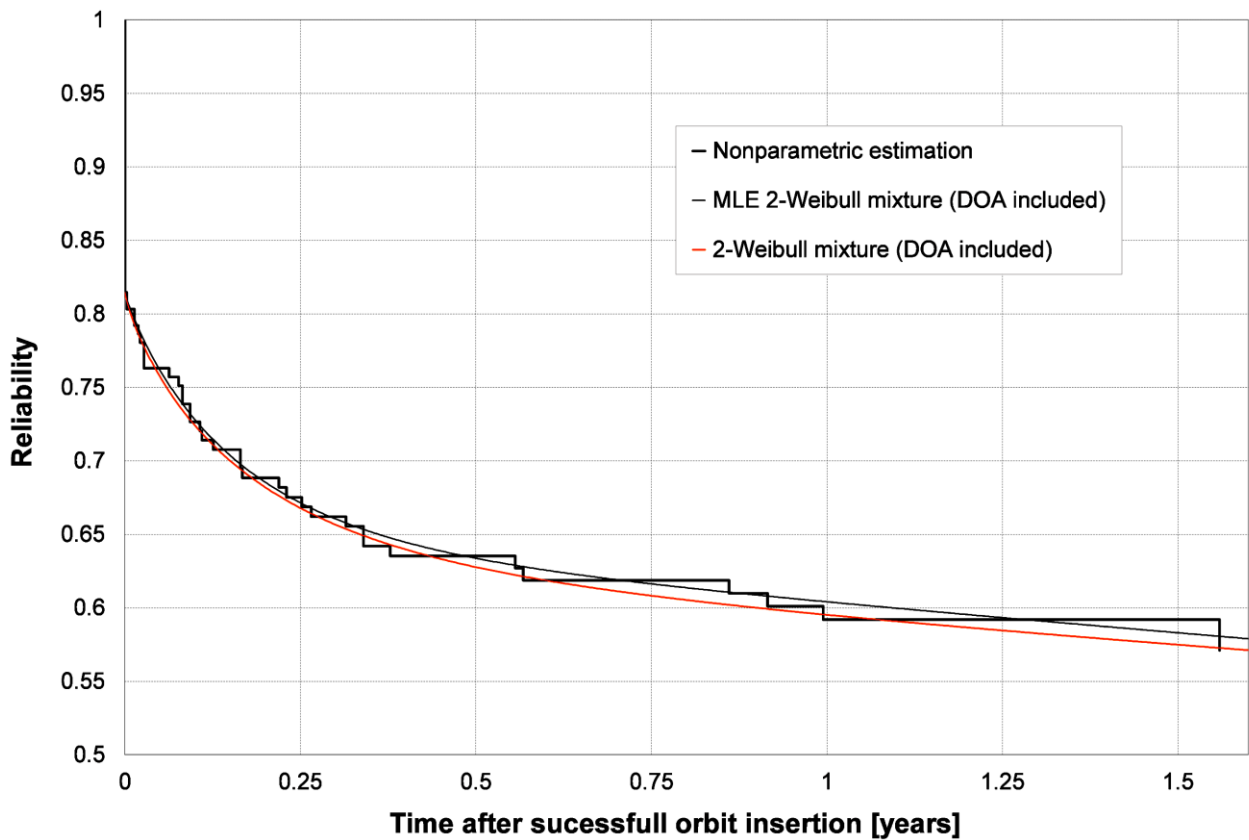


Figure 4-73: Comparison between the MLE (black) (equation (46)) and the nonlinear least-squares (orange) (equation (44)) fitted 2-Weibull mixture functions.

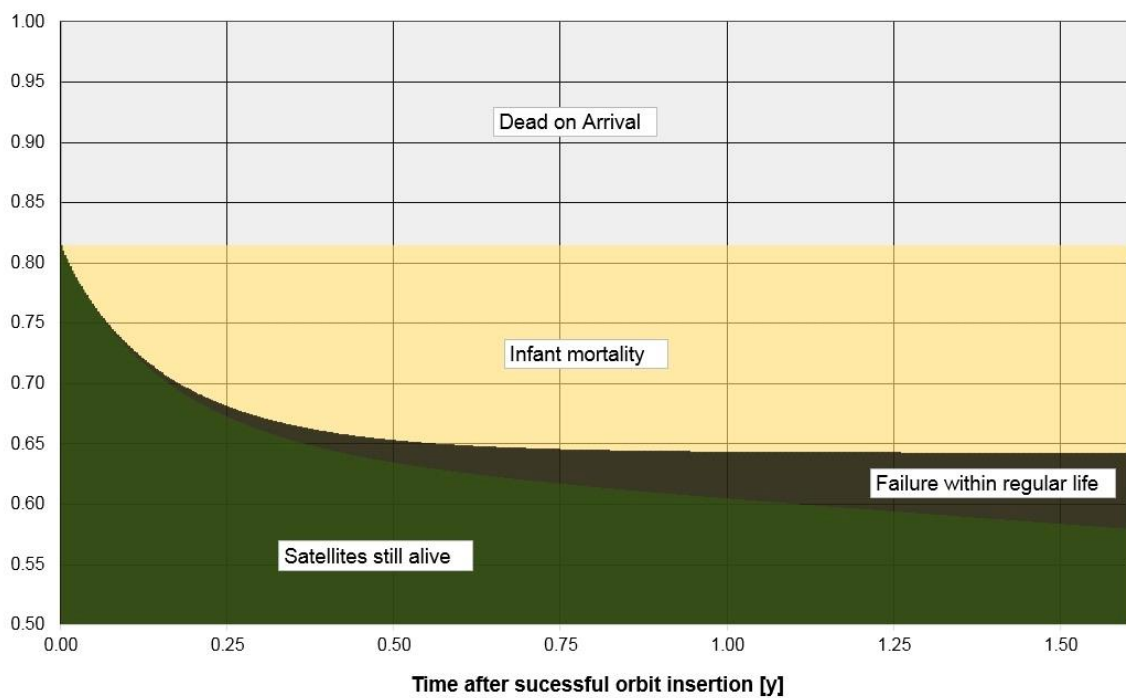


Figure 4-74: Fractions of CubeSats failed due to the three different portions of the MLE estimated PNZ-modified 2-Weibull mixture function (equation (46)) within their first 1.6 years on-orbit.

After assessing the overall system reliability of CubeSats, the nonparametric reliability of the involved subsystems was studied using data from the CFDB. For that purpose, the following 6 subsystems (plus an “unknown” category for failures, in which no specific subsystem was identified as a root cause) were defined: EPS, On-Board Computer (OBC), Communication System (incl. antennas) (COM), Attitude Determination and Control System (ADCS), Payload (PL), Structure & Deployables (other than antennas) (STR). The contributions of each subsystem to the satellite failures are depicted in Figure 4-75. Looking at data from larger satellites [117], the “unknown” category clearly strikes as they major source of error in early stages for CubeSats. While communication could not be established for many of the DOA satellites, interviews with CubeSat developers indicate that approximately half of the DOA cases are caused by the “unknown” category, while the developer had some indications of likely causes of DOA for the other 50%. The second largest contributor in the early phases and the largest one in later stages is the EPS, with more than 40% of all failures caused after 30 days (Figure 4-75). After 90 days, the communication subsystem accounts for nearly 30% of the failures. ADCS, PL and STR are contributing altogether less than 10% to the failure of the satellite. The three main subsystems causing CubeSat failures (OBC, EPS and COM) and the “unknown” category are modelled using nonparametric Kaplan-Meier estimation and parametric, PNZ-modified Single-Weibull fits, as shown in Figure 4-76 and Figure 4-77. As depicted in those figures, the subsystem-wise parametric function shows a bigger dispersion from the nonparametric data than the in overall reliability data. Thus, these parametric functions must be used with care.

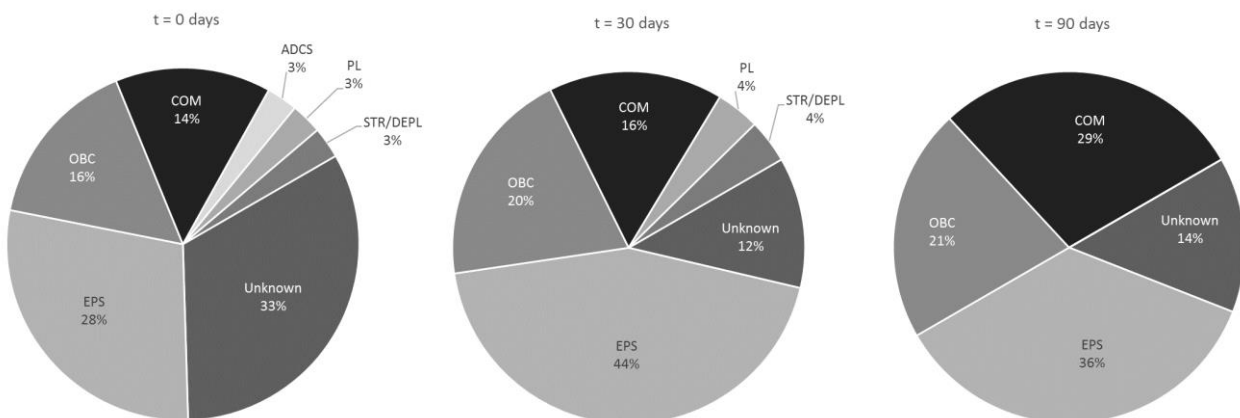


Figure 4-75: Subsystem contributions to CubeSat failure after ejection (incl. DOA), 30 days and 90 days.

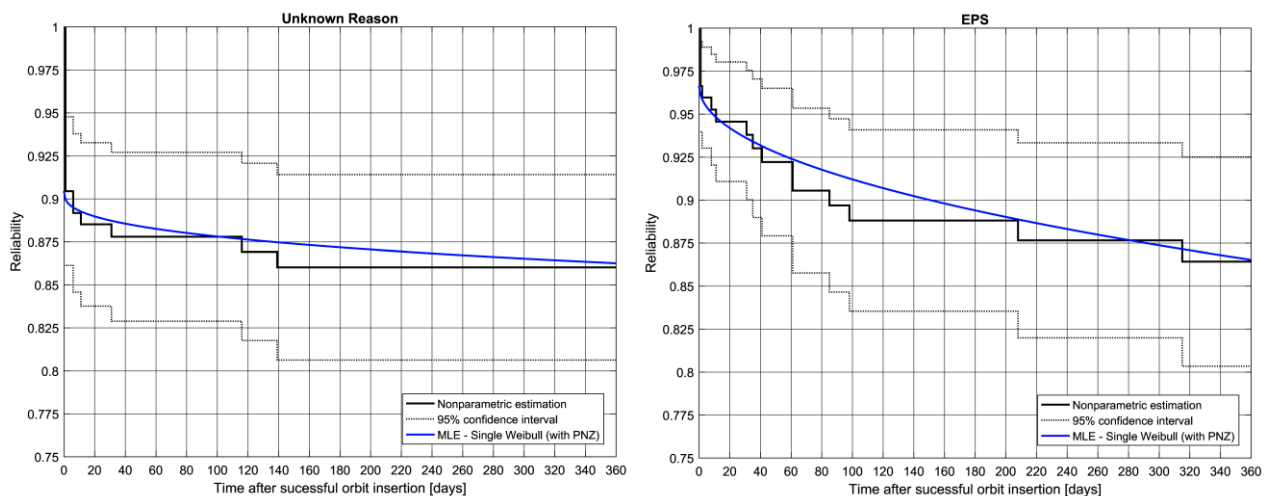


Figure 4-76: Nonparametric and Parametric Modelling of the “unknown” section and the EPS subsystem for a CubeSat failure during the first year on-orbit.

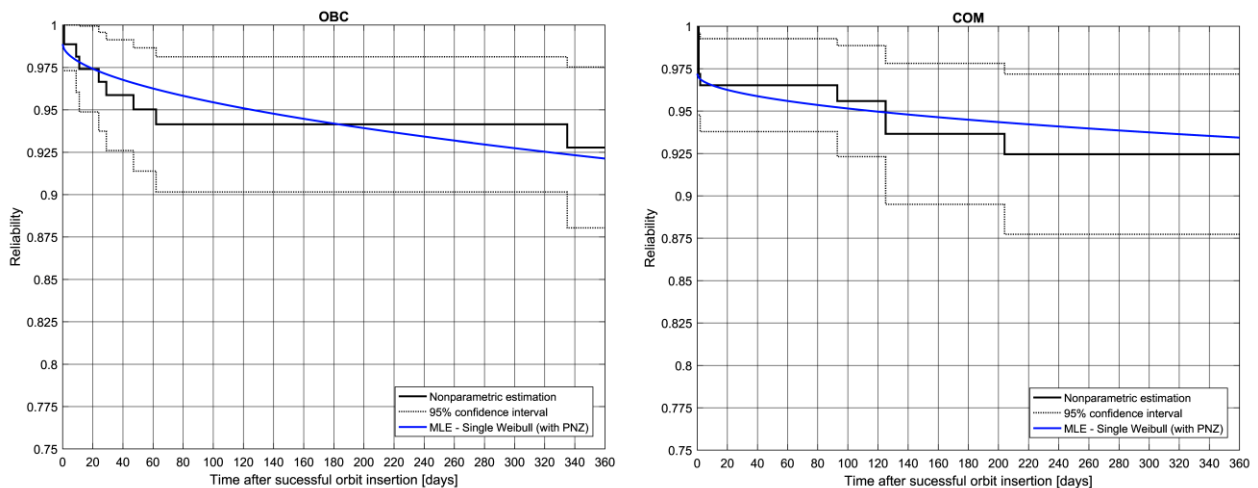


Figure 4-77: Nonparametric and Parametric Modelling of the OBC and the COM subsystem for a CubeSat failure during the first year on-orbit.

In addition to statistical data gathered for the CFDB, the survey conducted at the end of 2014 was also used to gain information on the developers' beliefs on the general reliability and specific reasons for failure of CubeSats. As aforementioned, of the surveys sent out to 987 CubeSat affiliated individuals, 113 were returned. Firstly, the likelihood of failure for a general, university-built CubeSat within the first 6 months was estimated by the group to be slightly below 50%, on average. A normal distribution was used to fit the expert elicitation data. Figure 4-78 shows the experts' judgement and the fitted normal distribution as red curve, with fitted parameters being $\mu = 48.98$ and $\sigma = 19.29$. For the first use, the normal distribution seemed a sufficient fit – nevertheless future work will be needed to estimate if there is a better fit on the experts' judgement. A second question was dealing with the expected likelihood of failure of the planned own CubeSat if the person was a team member of a to-be-launched CubeSat. A normal distribution was also used as a fit to the elicitation data. Out of 86 participants answering that part of the questionnaire, the normal distribution was fitted with $\mu = 16.53$ and $\sigma = 21.27$. Figure 4-78 depicts both, the judgement on the own CubeSat (blue) as well as the experts' opinion on a general, university-built CubeSat (red). The difference between the means of both normal fits is more than 32%, meaning that the estimation for the likelihood of failure for the own mission is rather optimistic or the judgement of other missions is very conservative.

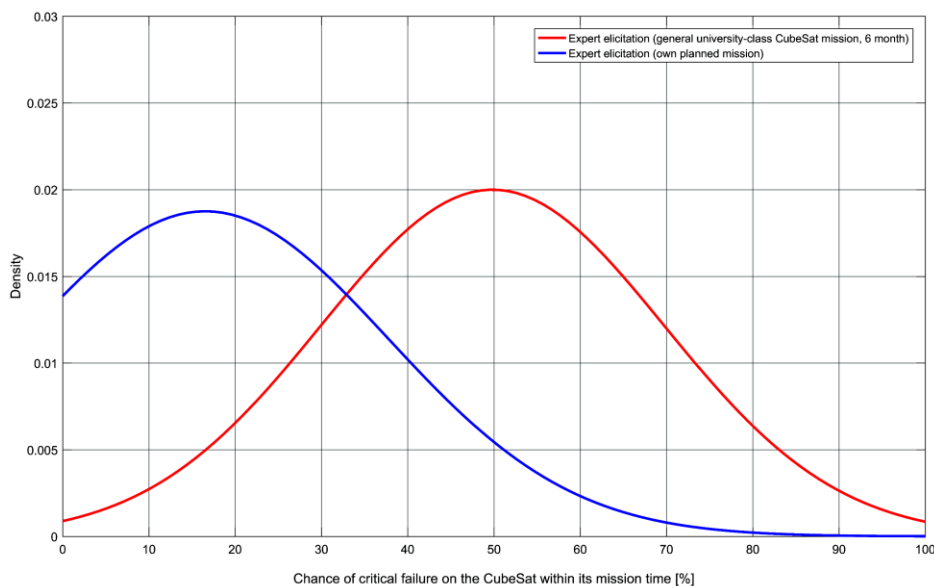
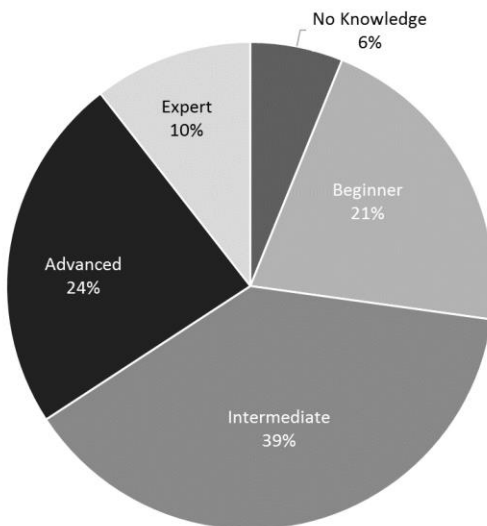


Figure 4-78: Developers' beliefs on the likelihood of failure for their own mission (blue) and on a general university-built CubeSat in its projected lifetime (86 developers).

Our survey also tried to gather information if the participants used failure or risk analysis on their satellite. As depicted in (Figure 4-79 left), 73% of the participants (114 answered that question) considered themselves not as a beginner or not as without knowledge in risk and failure analysis. Nevertheless, 34% of the group didn't use any method to quantify risk or reliability in the mission (Figure 4-79 right). For those who didn't use such methods, lack of time and lack of knowledge are the two biggest reasons not to implement them.

knowledge level of failure & risk analysis on satellites



applied Failure or Risk Analysis?

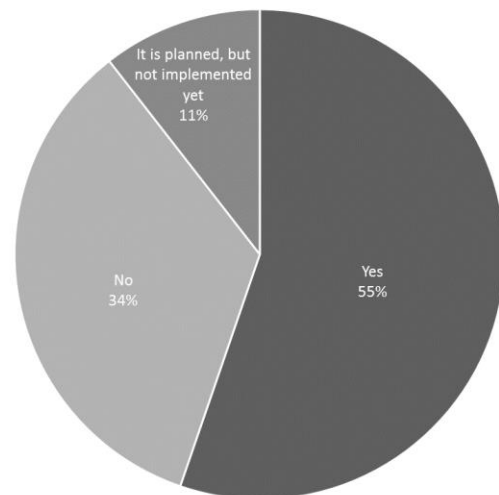


Figure 4-79: Survey results on knowledge level on risk & failure analysis on satellites (left) and survey results on the implementation of risk or failure analysis within their CubeSat program (right) (both 114 developers).

When then asked about the time spent in system level testing (or the time planned for that in case the satellite was not launched yet). The developers gave an average estimation of 616 hours when their satellite already had been launched (50 developers), but only estimated 255 hours if their satellite was still under development (42 developers). This more than double underestimation of time needed for system level testing might cause teams to run out of time at the end of their project, so they cannot conduct their system level test on a necessary scale, substantiating also the data of Swartwout [11]. When asked about the limitations of their planned system level tests, developers that had not launched their satellite estimated resources such as money and workforce (36%), access to test facilities (24%) and their schedule (24%) as the biggest limitations for their system level tests.

The same question was asked to developers that already had launched their satellite, with schedule now leading the list (53%), followed by resources (31%). For this group, access to test facilities played only a minor role (5%) for the duration of their system level tests. Looking more closely at the detailed reasons for limiting the system level tests, short implementation times in general ("The main factor was the short time to implement the mission, therefore several tests were skipped") launch opportunities ("Limited time between completing integration and ship out to launch site (literally a day)") and delay of subsystem development were amongst the reasons for reducing the system level test time. Although those factors are drivers for the schedule and might have multiple of underlying reasons, system level reliability test planning and assessment seems to be one possible solution, since beforehand knowledge of the time needed for system level testing could alter decisions made early in the projects.

To conclude this section, many of the CubeSats launched and built today are lost during their first phase of operations. The large percentage of DOAs and early failures is not acceptable if CubeSats should evolve into reliable and accepted platforms for scientific payloads and commercial applications. To stay attractive, CubeSats have to be launched and built fast, using appropriately selected COTS electronics and, due to

budgetary and time constraints, appropriately selected environmental test procedures that space agencies are using for their highly-reliable, expensive and large spacecraft. The solution to improve the DOA and infant mortality rate cannot be, in our opinion, to try to solve everything just with processes already used in the traditional space industry (e.g., reliability prediction or space-grade components). As we have seen in the earlier sections of this thesis, and as Swartwout pointed out, many of the early failures are due to poor system-level functional testing, i.e., the spacecraft was not operated (or not long enough operated) in a flight-equivalent state before launch [11]. Thus, many of the early failures could have been resolved by a certain amount of functional testing, rather than adding more and more complicated traditional acceptance and qualification tests. Despite their high rate of early failures, CubeSats changed the way how satellites are being built and how commercial and scientific missions can be carried out in the last decade. Their performance per mass figure of merit and fast delivery enables business models unthinkable of before their dawn. To further enhance their potential range of applications, the high rate of DOA and infant mortality has to be reduced in the near future.

As we have seen in Subsection 2.1.1, generally it is assumed when investigating failure free times or reliability of a product, that at $t = 0$ the system is free of defects and systematic failures [39]. For current CubeSats this is clearly not the case, and the additional high rate of early failures lead to the assumption that many CubeSats are currently in a region before useful life when there are launched. Amongst other reasons this could be caused by launch opportunities that cannot be missed (since CubeSats are always secondary payloads), limited knowledge, planning errors or drain of knowledge. Only rarely, failures of CubeSats are caused by random errors and this is also suggested by the little fraction of constant failure rate errors in our statistics. But as already pointed out, random errors are the main characteristic assumed by traditional reliability prediction methods. The overwhelmingly large fraction of CubeSat failures is caused by DOA and infant mortality, thus effects that have a decreasing failure rate over time⁷⁸. Thus, their reliability could be improved if we manage to bring those systems past their infant mortality on ground, and measure this by reliability growth modelling. To achieve that, we developed a reliability assessment method and tested it on our CubeSat, MOVE-II, which will be presented in the next section. This method is a supplement to environmental tests, not a substitute. Although the mostly used COTS components from automotive or industrial applications are not inherently unsuitable for space use, as we have seen in Subsection 2.3.1, the produced subsystems and systems must be qualified for the end-use environment⁷⁹. A further discussion about the data gathered on CubeSat reliability is presented in Chapter 5.

⁷⁸ Or in the case of DOA fail at $t = 0$.

⁷⁹ This means for us to withstand thermal-vacuum and the launch environment. We will later discuss the impact of radiation on CubeSats.

4.3 Reliability Assessment and Reliability Prediction of MOVE-II

In this section, we will present the applied methods to mitigate early failures on our CubeSat mission MOVE-II. In the first subsection the satellite itself and its development will be introduced. The focus will be not only on technical advancements, but also on approaches implemented that helped us to track failures, mitigate risk, and test our hardware more extensively throughout the development. As the author of this thesis is the project manager of the mission, some management lessons learned will also be shown. In the second subsection, the reliability growth model and the results of the reliability assessment of MOVE-II are presented. Those results are based on the methods of subsection one, so both sections will complement each other. Finally, in the third subsection, results of a reliability prediction model applied to our CubeSat are presented, mainly as an outlook into the future. As we have learned, reliability prediction assumes that the parts of a system work flawlessly together and thus the system itself is in its useful life when operated⁸⁰. Since this is not (yet) the case for CubeSats, any reliability prediction is a “best-case” assumption, and that is also true for our prediction. Nevertheless, the presented method could help in the future when different designs are traded-off against each other and the theoretical reliability of the system, subsystems or of parts shall be considered as one trade-off parameter.

4.3.1 The Development of MOVE-II

This subsection is an extended and adapted version of four conference papers ([244], [274], [275] and [276]) by the author of this thesis. Furthermore this subsection is partially based on results of the Master’s Thesis of Jonis Kiesbye [277] and the Interdisciplinary Project of Alexander Lill [278] both of them supervised by the author of this thesis.

In 2006, the LRT started the CubeSat program MOVE with the ambition of designing and building a 1U CubeSat verification platform, called First-MOVE (see Figure 4-80). The main goal of the program since then has been the hands-on education of undergraduate and graduate students. When First-MOVE was launched in late 2013, more than 70 students of different faculties had participated successfully in the project, with numerous educational and programmatic lessons learned. We operated First-MOVE successfully for one month, after which a major malfunction occurred in the satellites’ on-board computer, leaving the satellite in a mode in which it is only transmitting continuous wave (CW) beacons since then. Although the root-cause for this anomaly cannot be determined with absolute certainty, since two-way communication with the satellite was lost on that day, the strongest hypothesis assumes a data corruption in the magnetic read only memory (MRAM)-based boot sector of the satellite’s CDH system. Even though the Operating System of the satellite is not designed to write to the MRAM, an internal investigation concluded that memory-overlapping transients during the reboot-process of the OBC could be a possible source of MRAM data corruption. The short mission duration prevented several on-orbit mission objectives from being achieved. It was neither possible to obtain significant results from the solar cell experiment (primary payload objective) nor photos from the on-board camera (secondary payload objective). Nevertheless, the major in-house technology and spaceflight processing developments, culminating in the successful on-orbit operation of the self-developed subsystems, are sustainable results of the First-MOVE mission. From the beginning a slow, methodological and conservative approach was taken in early mission operations, slowly increasing the usage of the satellite’s functionality week by week as student operators became more familiar with systems on-orbit and on the ground. Therefore, most of the data obtained initially in one-way communication were sensor data coming from different subsystems and sensor locations within the satellite. This conservative approach, combined with the unexpected early failure of the CDH system prevented more aggressive two-way communication and payload commanding operations, thus reducing

⁸⁰ As we have seen, some reliability prediction methods consider infant mortality and/or wear-out, but this is always done on part level, not on system level.

the amount of data obtained from the satellite during its short mission duration. On the other hand, the risk involved in realizing a CubeSat mission for the first time and building major flight and ground subsystems in-house seem to make a conservative approach the better choice for university-based first-time teams. Furthermore, in the case of First-MOVE, this strategy enhanced the student involvement in mission-operations, resulting in the hands-on training of more than 20 additional students. After numerous attempts failed to recover the satellite, the end of First-MOVE's mission was officially declared on January 15th, 2014.

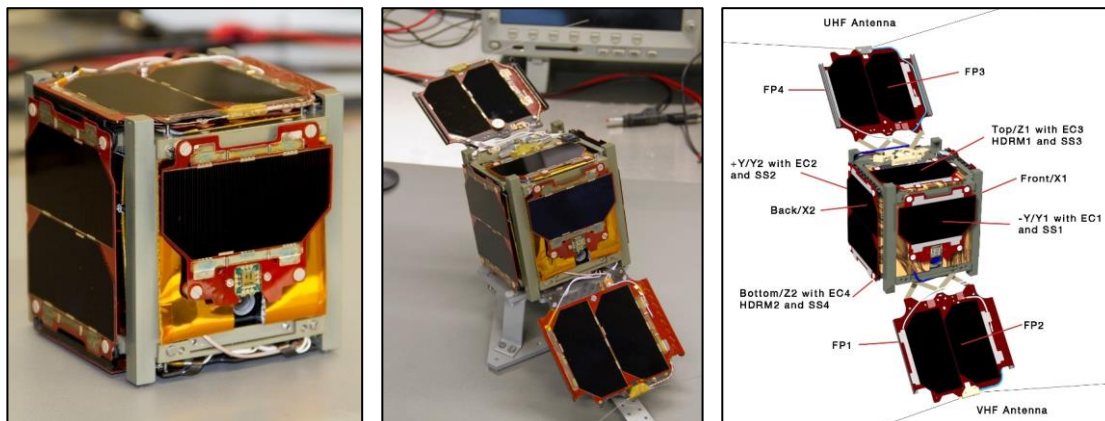


Figure 4-80: First-MOVE in launch configuration (left), the deployed configuration with both Flappanels (middle) and the nomenclature of the satellite and its sensor position (right).

Although the satellite passed a testing campaign both in LRT's thermal vacuum chamber as well as in the facilities of the Industrieanlagen-Betriebsgesellschaft mbH (IABG) near Munich, one lesson learned is to further extend in-house subsystem and integrated system-level testing of all components, including the purchased subsystems. Thermal cycling tests were important for the satellite, since a major, temperature-based issue on the latch-up protection unit was only found during those tests at extreme but realistic temperatures. Furthermore, the verification of the theoretical thermal simulation results was indispensable to avoid local overheating of components that had been covered with too much multi-layer insulation in the original design due to inexperience in space vacuum thermal design. The potential of overheating due to over-insulation was only discovered after more detailed thermal analyses were conducted by more experienced students and validated experimentally through actual thermal-vacuum tests. Despite the presence of a radio ground station on the Institute's roof and experience in mission operations, the MOVE team had to relearn and reestablish proper mission operations know-how for First-MOVE with the real satellite on-orbit. Acquiring the First-MOVE signal, and rapidly establishing data acquisition during the short overpass times, the low-cost amateur radios without complex and temperature-compensated frequency adjustment, posed initial challenges. On the other hand, these challenges provided invaluable lessons learned to all parties involved in the behavior of all system components and methods to compensate for hardware limitations. A full amateur radio operator certification program through volunteer mentors recruited from the local amateur radio community was established after First-MOVE at LRT. Vastly improved hands-on mission operations training for new personnel with extensive practical training on operational satellites from third parties is now implemented and highly recommended.

Other technical lessons learned in an academic environment without an established and externally imposed satellite design process included efficient process training for new students, from ESD, cleanroom etiquette and routine maintenance to operations training, such as regular lithium polymer battery management independent of vacation and exam schedule. Inexperience with the required battery management procedures, such as meticulous schedules for charge and discharge control resulted in multiple costly and dangerous lithium polymer battery malfunctions. The careful characterization of complex subsystems first through individual functional and performance tests under a variety of relevant environments, followed by a flat-sat style and increasingly integrated demonstrations and characterizations must happen early on and

with large scrutiny. Especially temperature-dependency of electronic circuits, from frequency, voltage and current shifts to change in timing constants must be understood and tested ideally in thermal vacuum environments or at least at the temperature extremes.

The necessity for longer, continuous operation tests of the fully integrated system is a lesson learned not only from the First-MOVE team but might also be major obstacle for other CubeSat teams worldwide, as described before. In hindsight, the overall testing time of both the major sub-systems and especially of the fully integrated system under relevant test conditions was insufficient to ensure reliable and successful operations. Since CubeSats are highly integrated systems, the careful planning of testability, integration and accessibility of all subsystems cannot be underestimated. In our opinion the mechanical integration of a CubeSat can be at least equally challenging as for larger satellite systems due to the extremely small volume and resulting tolerances of these satellites.

Programmatically, multiple launch delays between 2009 and 2013 were one of the biggest obstacles within the First-MOVE project. This extensive delay caused a significant knowledge drain due to the fluctuation and graduation of the involved shorter-term students and few longer-term staff members. This issue could only be addressed partially with written documentation through academic and project-relevant documentation. The need for project planning in all aspects of the product life cycle was underestimated at the beginning of the project. Several tests and (sub-) reviews were initially not deemed necessary because the satellite was “only” a CubeSat. The programmatic lesson learned here is that since a CubeSat is only slightly less complex than a larger satellite, CubeSat development projects need more or less the same number of technical reviews, despite the small size of the vessel.

The educational aspects while using a CubeSat for teaching purposes in a university environment include a) planning the project around students’ academic schedules rather than in a traditional, linear fashion, and b) the careful selection and assignment of team members to subsystem teams to retain student motivation and an even distribution of more and less experienced members. A key lesson was learned concerning the previously mentioned drain of knowledge about the satellite or its subsystems due to students leaving the project, having either completed their thesis work or their respective graduate or undergraduate programs. To remedy this situation, which many university-based CubeSat teams experience, interested students should not just be assigned to a specific thesis topic, but also encouraged to stay involved in the project before the beginning and beyond the duration of their thesis work. Since retention of specific students over longer periods of time is very difficult in the German academic curriculum, which favors diverse projects over specialization, a voluntary, dedication-based approach through membership in a student association was chosen for MOVE-II to improve both academic success and retention of knowledge.

The need for the inclusion of external experts was another important lesson learned from First-MOVE. Technical hands-on education of students in different subsystems of a satellite can be drastically enhanced by the expert knowledge of senior engineers in the field. Although there are some management resources needed for the identification and coordination of those voluntary experts, the possible benefits of expert knowledge transfer from experienced professionals outweigh the costs. Besides the traditional approach of involving external experts during sporadic major design reviews in the project, there was an even bigger demand identified for more frequent and continuous external feedback from aerospace professionals as easy accessible technical mentors. This would have been especially helpful during the critical subsystem design phases leading up to the major reviews. In total, more than 70 students successfully gained hands-on experience through the First-MOVE project that would have otherwise been not obtainable via the traditional curriculum. In addition to the obvious aspect of working in a team, students learned the real life, hands-on work on a satellite project, including reviews, milestones and deliverables – aspects that are common for projects in the aerospace industry. Another educational aspect is the fact that students had to learn to plan their own personal work schedule and the amount of work they commit to doing; valuable skills for their professional careers. Finally, the students not only benefited from doing things “right”, but also from sometimes making mistakes and successfully recovering from setbacks. This personal

experience of failure and recovery cannot be created by a traditional academic curriculum and the lessons learned from it are engrained deeper into the student's memories than any lecture ever could.

Since April 2015, we are continuing the hands-on student education at LRT with our second CubeSat, called MOVE-II. With more than 150 students involved so far, MOVE-II is currently awaiting shipment to the launch provider, with its launch scheduled for autumn 2018. MOVE-II is a 1.2 kg, 1U CubeSat. Being limited to single-unit size during launch, the satellite, as its predecessor, uses deployable solar panels to overcome the power limitations of the single-unit envelope (Figure 4-81).

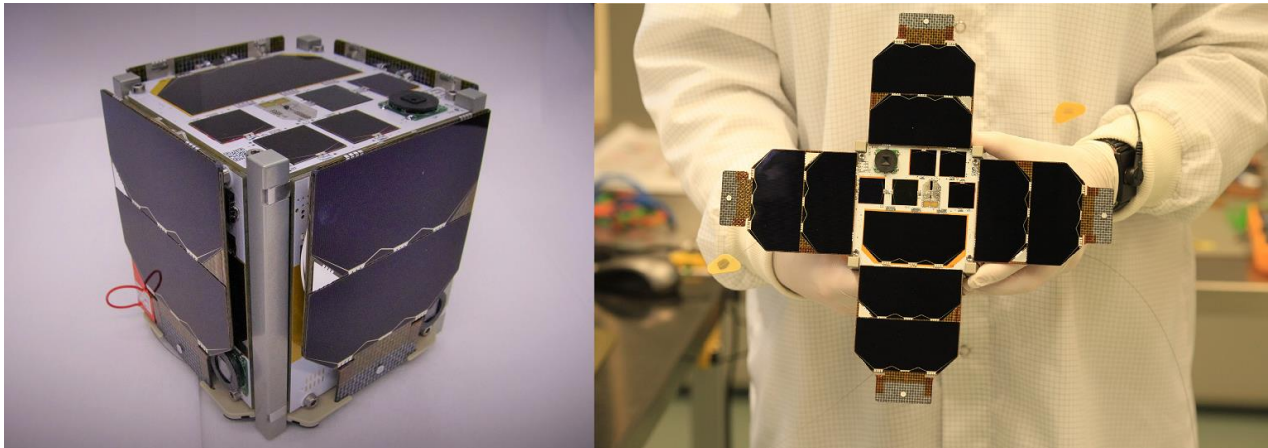


Figure 4-81: MOVE-II in launch configuration (left) and deployed configuration (right).

The four deployable solar panels are of carbon fiber reinforced plastics and are equipped with two 4-junction solar cells each. A reusable shape memory mechanism holds down the panels during launch and releases them after deployment. The mechanism, already tested on a sounding rocket [279], allows repeated tests of the mechanism and thus a true TLYF philosophy. In the following, we will explain the satellites' main subsystems, which are also depicted in the explosion drawing of Figure 4-82.

The core of the satellite is an electronic stack of six printed circuit boards (PCBs), plugged together using PC/104 sockets. The so-called Toppanel, which houses the payload of the mission, is an additional seventh PCB stacked on top of the electronic stack, and it is connected to the stack via an adapter board. The Toppanel houses one full size solar cell (8 x 4 cm) and four corresponding isotype solar cells (each 2 x 2 cm) as the PL of the mission. As its scientific goal, the MOVE-II CubeSat will be used for the verification of these novel 4-junction solar cells under space conditions. An additional isotope cell (2 x 2 cm), located also on the Toppanel, will be flown without a cover glass to study the resulting accelerated degradation process. Figure 4-83 shows the circuitry of the PL on MOVE-II, which measures the current-voltage curve of each solar cell. For these measurements, all solar cells are connected for 4-wire sensing. The voltage at the solar cell is measured between the contacts and the actual current of the cell is measured as a voltage drop across a shunt resistor. For the sweep of the current-voltage curve, a metal-oxide-semiconductor field-effect transistor (MOSFET) is added to the circuit and used as a variable load. On the backside of the PL board, temperature sensors are attached behind each solar cell. Furthermore, a sun sensor evaluates the actual sun angle seen by the solar cells.

For the verification of the PL measurements, various tests were conducted. The accuracy of the measuring circuit on the final flight model was evaluated with a Keithley 2400 sourcemeter. The observed uncertainties were less than $\pm 0.1\%$ for the voltage measurement and less than $\pm 1\%$ for the current measurement. A significant deviation of the stated accuracy can be evaluated with a self-test of the PL, which is regularly performed before the start of the measurements. The setup proved functional and measured reliably in vacuum under the expected illumination in space. Furthermore, all components of the measuring circuit are

radiation tolerant and should not exhibit degradation effects before 10 krad. More information on the PL of MOVE-II can be found in a conference paper [280], co-authored by the author of this thesis.

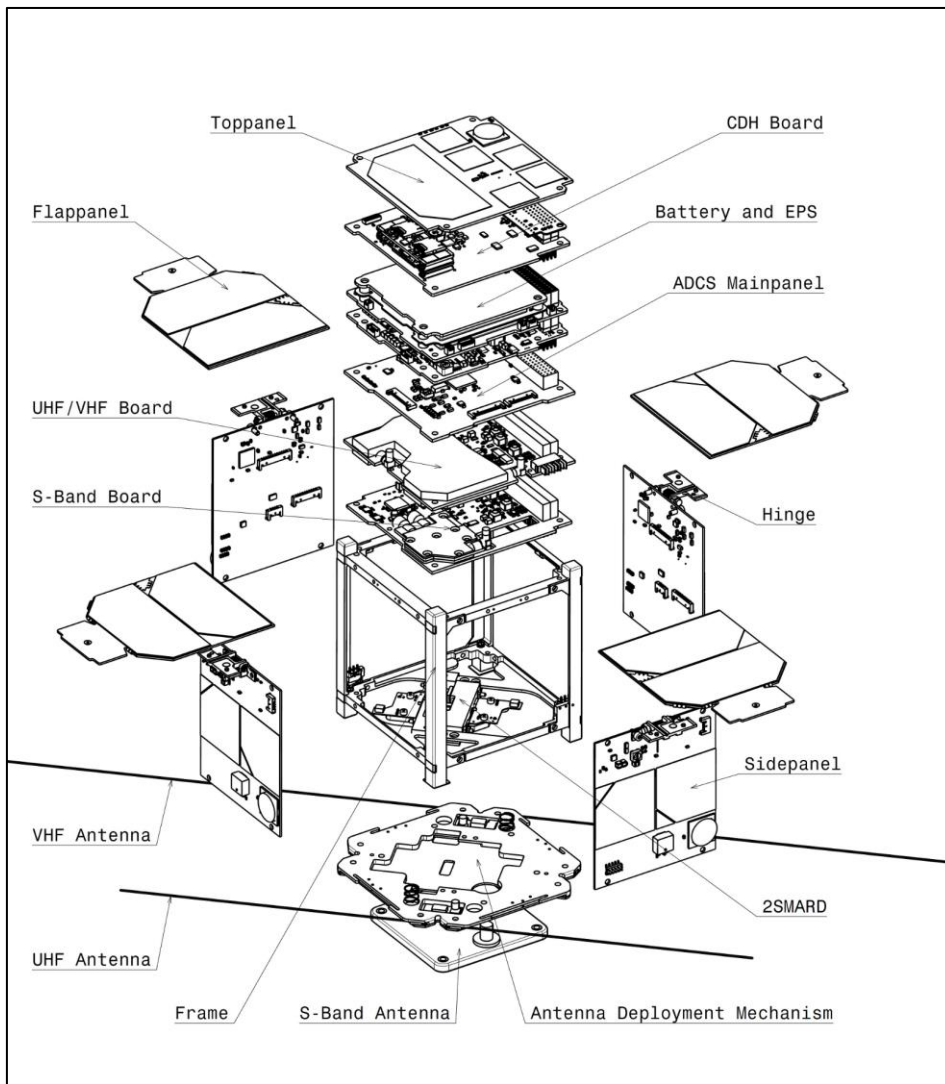


Figure 4-82: Explosion Drawing of MOVE-II.

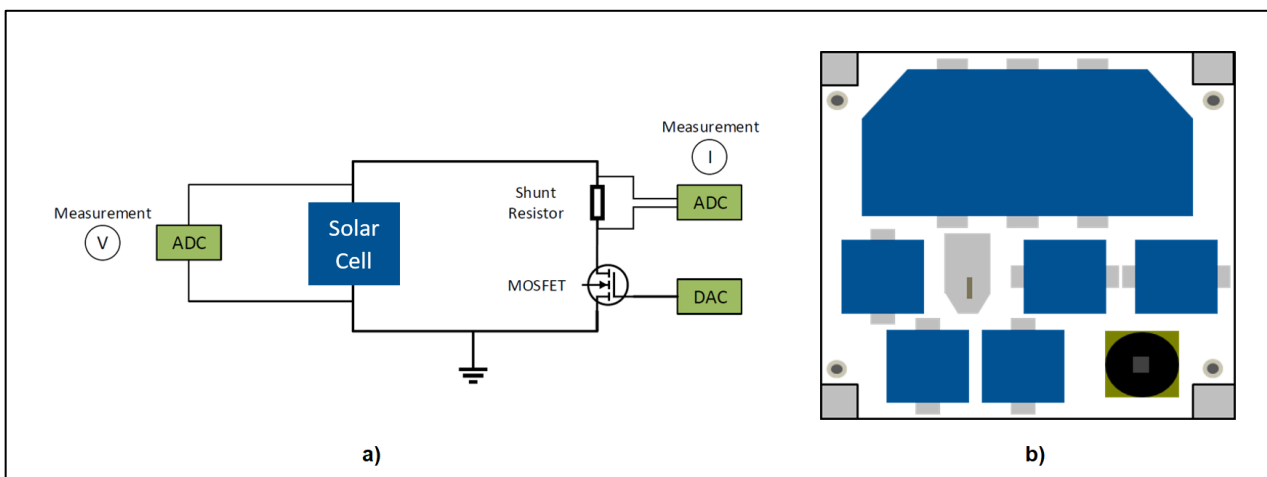


Figure 4-83: (a) Circuit for the measurement of the current-voltage curve of solar cells on MOVE-II. (b) Top side of the CubeSat with solar cells of the payload and one sun sensor

Besides the PL, the Toppanel also hosts electronics and actuators for the ADCS. Thus, the Toppanel is also part of the ADCS system. MOVE-II will be the first CubeSat of TUM utilizing a magnetorquer based, active ADCS. The ADCS consists of five Printed-Circuit-Boards with directly integrated magnetic coils, forming the outer shell of the spacecraft (the so-called Sidepanels being the other four besides the Toppanel), and the so-called ADCS Mainpanel, located in the middle of the board stack of the satellite (see Figure 4-84).

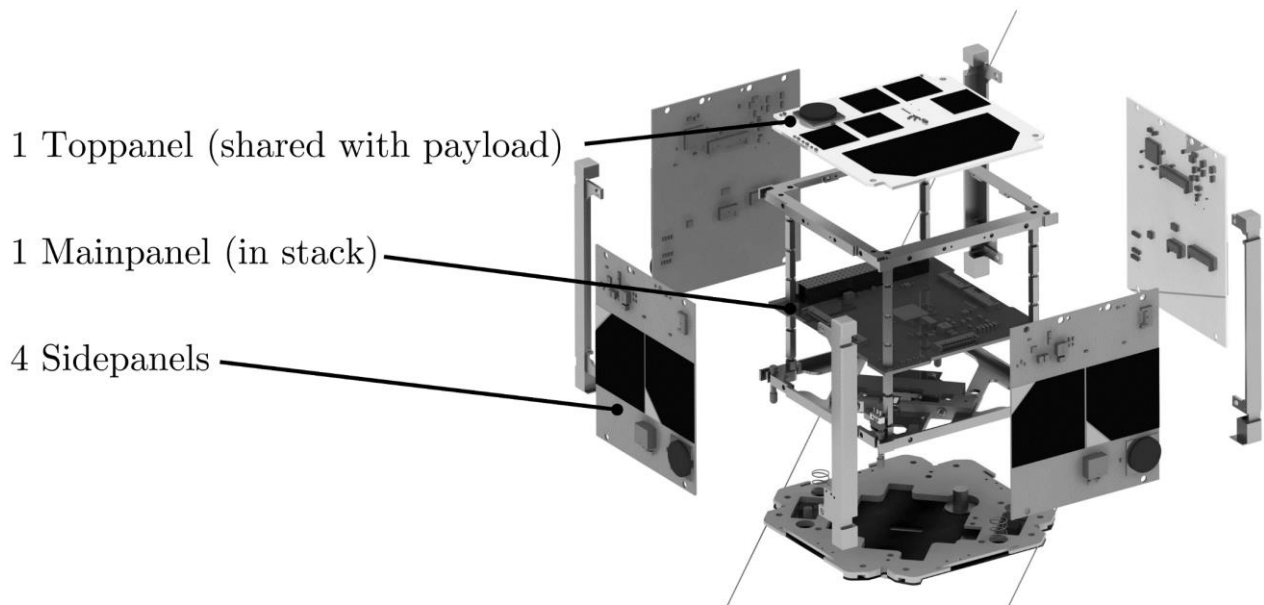


Figure 4-84: Toppanel, Mainpanel and Sidepanels of the ADCS on the MOVE-II CubeSat.

The ADCS Mainpanel controls the current of the magnetic coils of the other five panels and of its own, sixth coil. It is the central unit of the ADCS hardware and is in the board stack and connected to the remaining satellite by a PC/104 interface. Each Sidepanel has its own microcontroller and is connected to the ADCS Mainboard with one of two redundant Serial Peripheral Interface (SPI) buses. Each panel features a PCB integrated magnetorquer coil, a corresponding coil driver, a microcontroller, a three-axis magnetometer, and a three-axis gyroscope. In addition, all Sidepanels and the Toppanel are equipped with a sun sensor. The measurements of the sensors are independently acquired and pre-processed on each panel. These data are accumulated on the Mainpanel and used for the computation of the attitude determination and control algorithms.

The MOVE-II mission requires two main control strategies on which the design of the ADCS system is based: detumbling and sun pointing. After ejection from its deployer, the satellite may experience undesired high angular velocities. A simple and robust B-dot controller, which has been implemented on several CubeSat missions before, is used to slow down the rotation of the satellite, which is called detumbling. After successful detumbling, the top side of the satellite is directed towards the sun to ensure scientific operation by using a linear model-based control approach. Moreover, an Extended Kalman Filter (EKF) is implemented to estimate the satellite's attitude by using a provided set of sensors. Investigating the system functionality and performance requires appropriate testing environments. The challenges of testing our ADCS yield to the development of an ADCS prototype and a Hardware-in-the-Loop (HiL) setup. They are presented in detail later in this subsection as an example of tests conducted in MOVE-II on subsystem level. More information on the ADCS system of MOVE-II and the control strategies implemented are presented in two conference papers [281] [282], both co-authored by the author of this thesis.

The MOVE-II COM consists of two independent systems: a continuously operating Ultra high frequency (UHF)/Very high frequency (VHF) system (see Figure 4-85 left) and an auxiliary S-Band system (see Figure 4-85 right). The UHF/VHF system provides an attitude independent, highly available link. This is the main COM link used to bi-directionally transmit all data, including TT&C. This link has a data rate of up to 25 kb/s in both directions. The antennas of the UHF/VHF transceiver are hold down and released by the redundant shape memory mechanism, which will be discussed later in more detail. For higher bandwidth, the S-Band system can provide up to 3 Mb/s additional downlink data rate and 150 kb/s on the uplink using a patch antenna located at the bottom of the satellite.

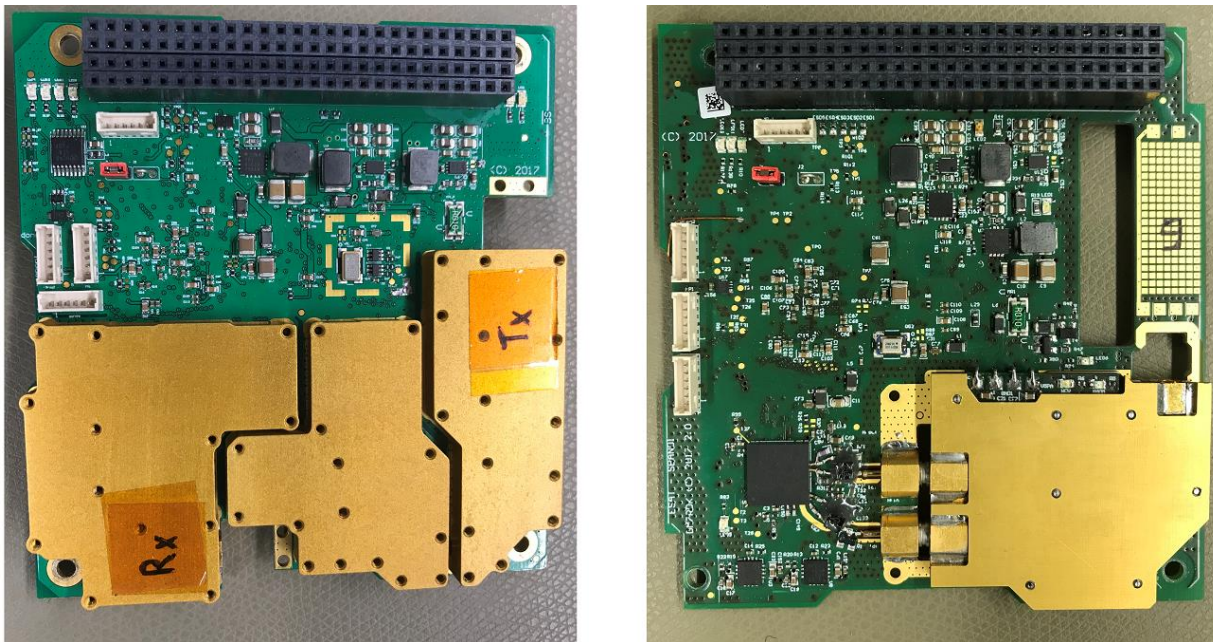


Figure 4-85: UHF/VHF Transceiver (left) and S-Band Transceiver (right) of MOVE-II.

The student-developed layer 2 protocol Nanolink provides quality of service features for user applications on both communication channels. This includes guaranteed data rates for different streams and use cases. Additionally, an automatic repeat request protocol ensures delivery and improves link quality for upper layer protocols. Nanolink is specifically tailored for moderate signal quality and efficiency in low bandwidth-delay applications. More details on Nanolink can be found in a conference paper [283], co-authored by the author of this thesis.

The design goal of the physical layer is to maximize data rate while retaining a comfortable link margin. The limiting factor in UHF/VHF is bandwidth, and power in S-Band, respectively. The resulting design uses phase-shift keying modulation, and powerful Accumulate, Repeat-by-4, and Jagged Accumulate (AR4JA) Low-Density Parity-Check (LDPC) codes by the Committee for Space Data Systems (CCSDS) on the downlink. The parameters of the system are adapted to fit the requirements of the respective link. It is possible to change these parameters at runtime since the whole signal processing is implemented within an FPGA, which offers the possibility for highly parallel, complex signal processing and advanced coding schemes. The digital signal processing within the FPGA utilizes complex I/Q samples, which are exchanged with dedicated radio frequency hardware. This enables the use of virtually any modulation on the up- and downlink. The FPGA image can be reprogrammed on-orbit, allowing to change the modulation and coding arbitrarily. To achieve a small footprint, the design relies on highly integrated components. One focus during selection of the components were power consumption and reliability. As noted before, automotive grade COTS components are qualified for a wider temperature range than consumer grade components, thus automotive grade components were mainly used for both transceivers. For key components the radiation tolerance was also taken into account. An inherently radiation tolerant MRAM was selected for the FPGAs

configuration storage. Both transceivers were validated as part of the TDP-3 Vanguard experiment on the Balloon Experiments for University Students (BEXUS) 22 mission in October 2016. During this mission the communication link was tested in the stratosphere over a distance of 270 km. More information on this long duration test can be found in [284], co-authored by the author of this thesis.

The EPS and CDH are both subsystems that were bought off-the-shelf from 3rd parties. Thus, the hardware and electronics of these systems will not be described in here. On the software side of the satellite, an overview is depicted in Figure 4-86. In general, a more exhaustive description of the satellite can be found in the MOVE-II System Documentation [285].

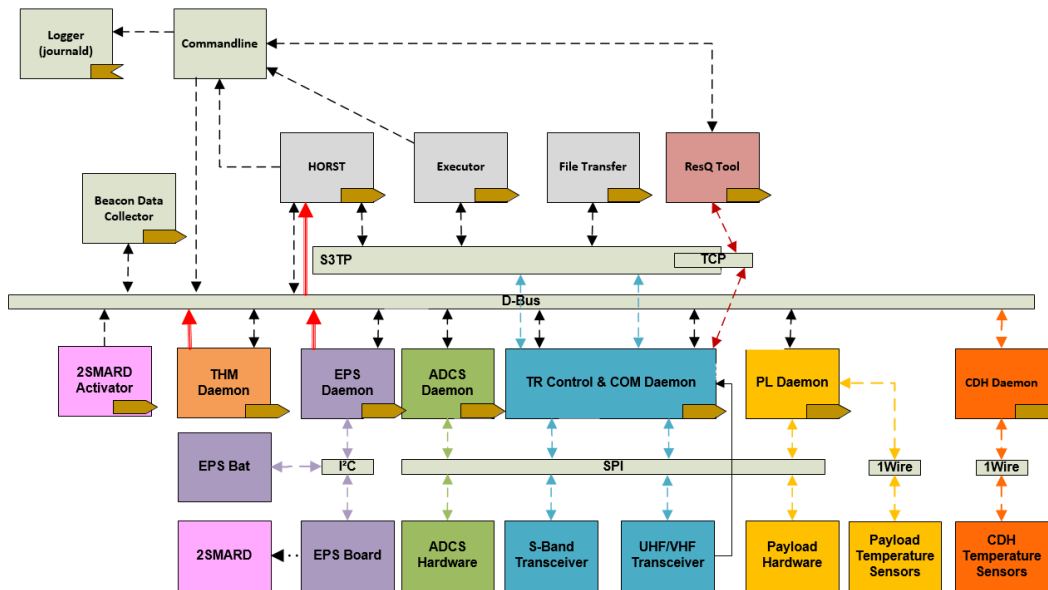


Figure 4-86: Overview of the Software and Interfaces of MOVE-II. Source: MOVE-II System Documentation [285].

The structure of the satellite was designed to ease access to the stack, as can be seen in Figure 4-87. The guiding rails of the CubeSat can be attached after the stack was integrated. The usual power restrictions of the 1U envelope are overcome by four deployable solar panels, which are held down and released by a reusable Shape Memory Alloy Hold-Down & Release Mechanism, called 2SMARD. This allows repeated tests of the mechanism and true TLYF philosophy.

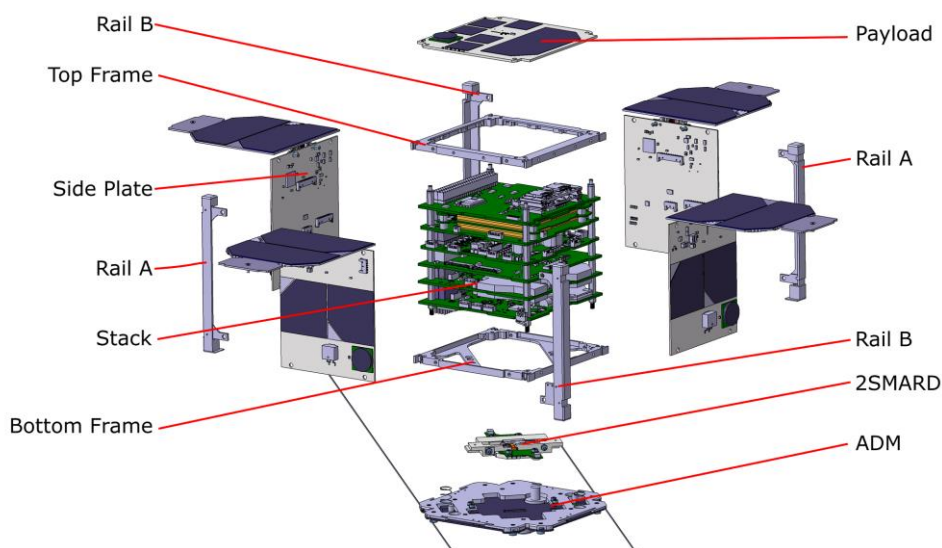


Figure 4-87: Structure of MOVE-II. Image Source: MOVE-II System Documentation [285].

2SMARD not only opens the deployable solar panels, it also releases the so-called Antenna Deployment Mechanism (ADM), which houses both the UHF and VHF antennas of the satellite. As soon as the deployment is triggered in space, two mechanical springs push the ADM away from the satellite's body, freeing the antennas. 2SMARD, depicted in Figure 4-88, is a redundant mechanism in which two sliders are used to hold-down the ADM to MOVE-II's body (and through the ADM also the Sidepanels are held down). When the mechanism is activated, current flows through the shape memory alloy (SMA) springs, leading to a contraction of them and a subsequent opening of the mechanism (see Figure 4-88 right). Only one slider has to move to free the cassette.

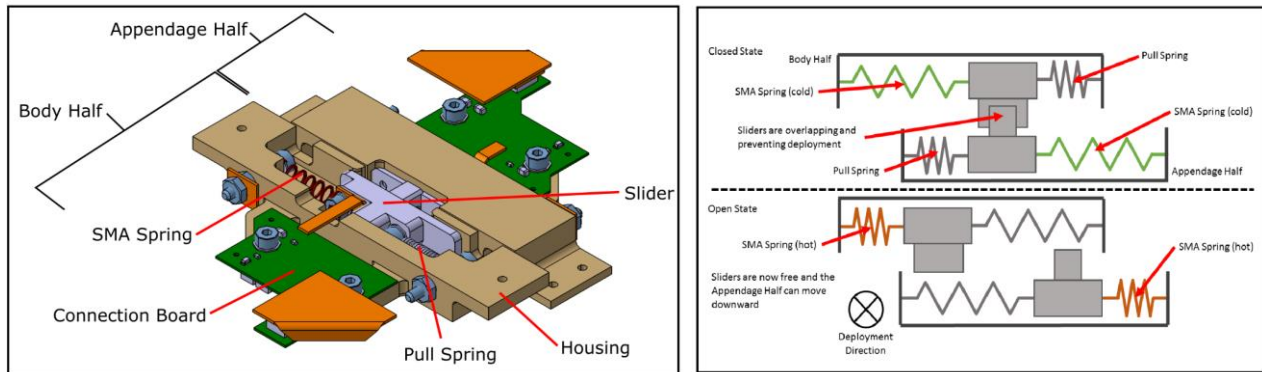


Figure 4-88: Overview of 2SMARD (left) and working principle of 2SMARD (right) of MOVE-II. Image Source: MOVE-II System Documentation [285].

From the beginning, we tried to apply the lessons learned of First-MOVE and other CubeSat teams in the program: We focused on testing and careful characterization of complex subsystems, similar to the Bread-Brass-Silver-Gold approach of the Air Force Research Laboratory's (AFRL) University Nanosatellite Program [286] [287], and built prototypes and brass-boards often and early. Due to recent advancements in additive manufacturing as well as custom-made thermal prototypes of all subsystems, we were able to conduct individual functional and performance tests under a variety of relevant environments early on, followed by a long phase of flat-sat style and integrated system level tests.

As MOVE-II was designed with a high power consumption due to two transceiver boards and the six coils build in the Sidepanels for attitude control of the satellite, a prototype that represented the overall thermal balance of the system was built early in the design phase. The thermal prototype (Figure 4-89) consisted of an aluminum frame from an earlier design iteration and several two-layered PCB's representing each board of the different subsystems. To simulate the heat loads on the boards, heating foils and resistors were mounted on the correspondent places of the future heat dissipation on the PCBs. The prototype was then put into LRT's Thermal-Vacuum Chamber (TVAC) to determine which board stacking order provides the least amount of thermal stress on the battery, the most critical system with the most stringent operational temperature limits (between -10°C and $+50^{\circ}\text{C}$).



Figure 4-89: Thermal Prototype of MOVE-II and TVAC of LRT.

After some iterations the best stacking order with regards to the temperature of the battery was determined. It resulted in the S-Band transceiver, the one with the most heat dissipation, being the lowest board, followed by the UHF/VHF transceiver, the ADCS Mainpanel, the EPS board with the battery stack and finally the CDH board on top of the PCB stack. The ADCS Mainpanel and the CDH board dissipate less heat and thus serve as an insulation from the high heat dissipation of the UHF/VHF transceiver and the Toppanel, which is, ideally, in direct sunlight most of the time due the sun-pointing requirement of MOVE-II. Besides the determination of the best stacking order, testing of incoming subsystem PCBs was one of the major tasks of THM, resulting from lessons learned of First-MOVE. As PCB hardware arrived, it was put into the TVAC and equipped with external thermocouples to detect faulty craftsmanship and to get an impression of its thermal behavior in vacuum (see Figure 4-90 left). Also, since MOVE-II, as many other CubeSats, relies on automotive and industrial grade COTS electronics, a qualification of subsystems in TV is necessary and can be also seen as a burn-in test of parts (and subsystem level if the whole subsystem was purchased from a 3rd party).

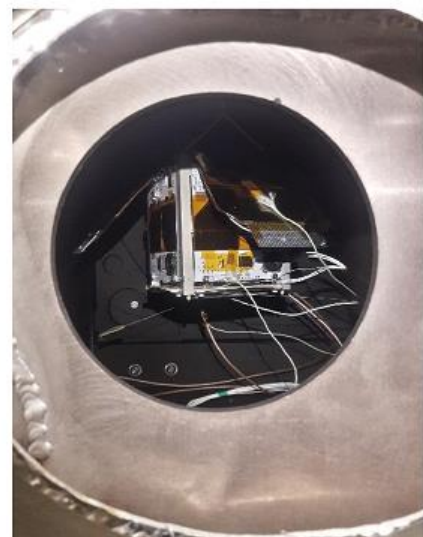


Figure 4-90: Subsystem-level test in the TVAC (left) and System Level testing of the MOVE-II FM (right). Image Source: [288].

With the acquired knowledge during the TV tests using the thermal prototype, the initial ESATAN model, which represented coarsely the geometry of MOVE-II and material property values found in literature, was refined. Thermal conduction through the PC/104 and the standoffs could be adjusted, and optical and

material properties corrected. After testing the subsystem hardware in the TV, the initial power budget was updated, and the changes implemented in the ESATAN model. Several orbital simulations according to the concept of operation on-orbit were performed. These simulations calculated battery temperatures that were close to the upper limit of the operational temperature. To lower the battery temperature additional simulations with a white coating on the Sidepanels (initially green) were conducted. The results showed that just by changing the optical properties of the Sidepanels, the battery temperature was reduced by around 12°C, thus increasing the margin to the upper operational temperature threshold.

Thermal balance tests on the integrated EM provided new insights regarding material properties, to thermal coupling between subsystems and structure, and to the thermal behavior of the whole satellite. This information was used for the correlation of the ESATAN model. The geometry was refined by adding the cages of the transceiver boards and adjusting the mesh of some boards for a better representation of hot spots. An overall accuracy of $\pm 10^\circ\text{C}$ was achieved. The most critical part, the battery, achieved in the worst cold case (assuming no internal heat dissipation) and the worst hot case (active uplink on the UHF/VHF transceiver) temperatures of 1.8°C and 25°C. Thus, the battery is still within the operational temperature limits by a safety margin of $\pm 10^\circ\text{C}$. The simulated temperature distribution during an overpass with an active UHF/VHF uplink can be seen Figure 4-91.

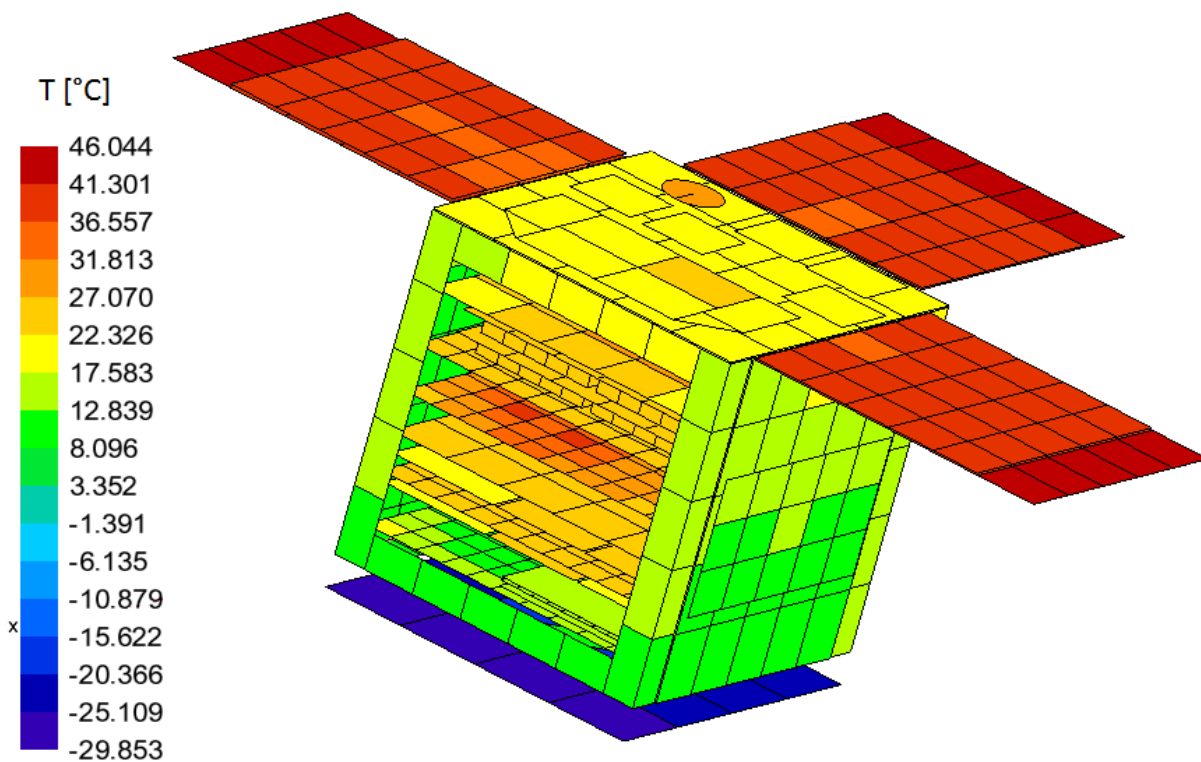


Figure 4-91: Calculated temperature distribution with an active UHF/VHF transceiver uplink of the MOVE-II ESATAN model with inner board stack.

Finally, both the EM and the FM (both models depicted in Figure 4-92) were built and tested in the laboratory and in the TVAC. In the TVAC, thermal balance tests and thermal cycling tests were conducted. For the thermal balance tests different operating modes of the satellite were active until each mode reached a thermal steady state. These tests were conducted at a warm as well as a cold environmental temperature. Thermal cycling was done to expose the satellite to thermal stress and find faulty craftsmanship. Hereby

several hot cycles, with maximum heat dissipation on the satellite, and several cold cycles, with minimum heat dissipation on the satellite, were performed.

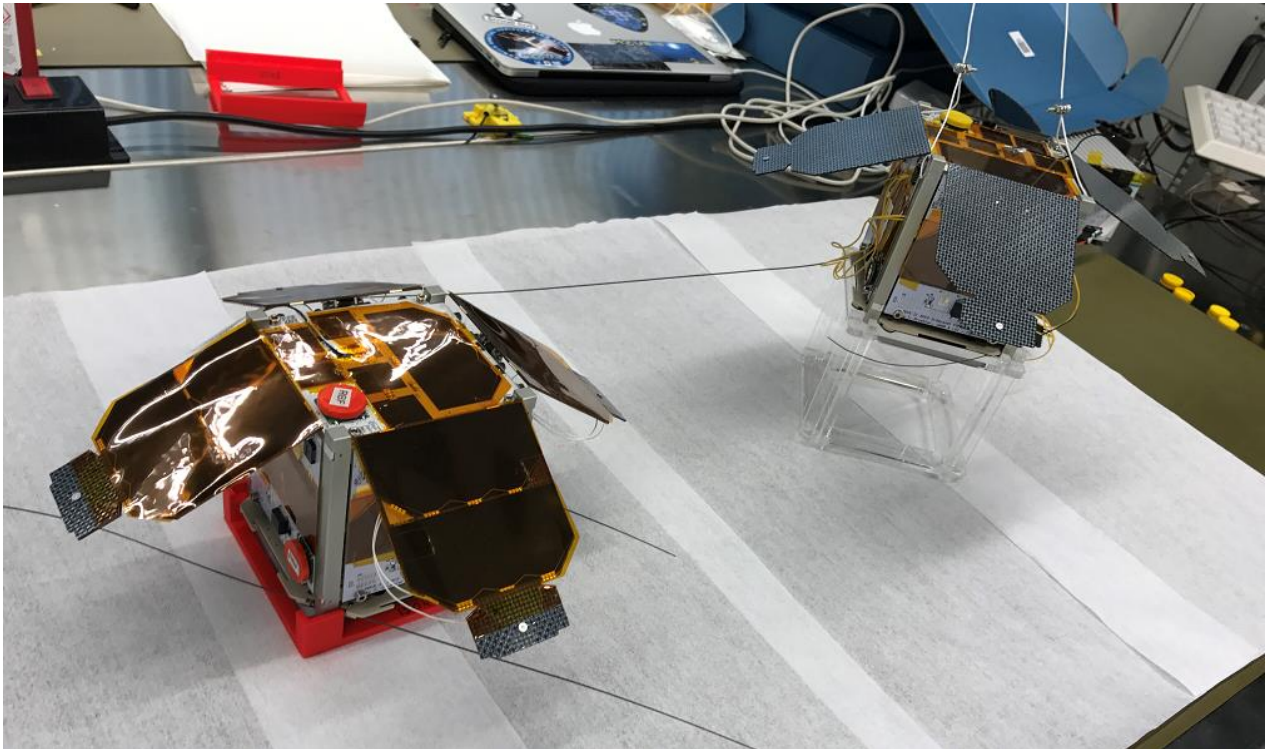


Figure 4-92: MOVE-II FM (left) and EM (right) in the cleanroom of LRT. Solar cells and sun sensors of both models are covered for protection.

As we have seen in Chapter 4.2, the success story of CubeSats is currently jeopardized by many DOA and infant mortality cases. Applying the lessons learned of First-MOVE and other CubeSats, but also using methods from terrestrial applications, we tried to reduce the risk of facing early failure in MOVE-II. In the following, we will report on selected methods, applicable also for other CubeSat teams: Additive manufacturing can help mitigate integration errors and can provide fast access to ground support equipment (GSE). We chose Agile Software Development as the most suitable method for developing our CDH and Mission Operations software, since we wanted to not only test our hardware but also the software early and rigorously. A hardware testbed for ADCS as well as an ADCS HiL environment are examples of low-cost test equipment, based on open-source single-board computers that were created to test subsystems in a TLYF manner. To ensure reliable and successful operations of MOVE-II, we focus on system level tests, carefully monitor the bugs and errors on system level, and evaluate the remaining testing time needed and the number of not detected bugs in the system. We will describe the overall test setup for system level tests and the FRACAS built for MOVE-II in this subsection and focus on reliability growth modelling, based on results of system level tests in the next subsection. As already pointed out in chapter 3, the main goal of this thesis is to increase the reliability of a university-built satellite, thus all methods are mainly applicable in a university environment. Nevertheless, as university-built CubeSats are just more restricted cases (resources, knowledge, schedule, volume, etc.) of traditional missions, some solutions might be also of use for larger missions.

We manufactured a 3D printed structural model of MOVE-II very early in the design process (see Figure 4-93) to shift integration risk upfront. Lessons learned from First-MOVE as well from other teams showed that the first assembly of final hardware often ended in mechanical incompatibilities. Thus, the intentional purpose of this prototype was to prove that our satellite can be integrated and to determine the final cable paths. Being a mechanical representation of satellite, it had no electrical functionality. All boards were

produced with the Fused Deposition Modeling (FDM) technology and some components on the boards were implemented by plastic or wooden blocks glued on the boards. We also produced a prototype of the frame by laser-sintering, as FDM did not yield the required accuracy. The Side- and Flappanels were simply cut out of polymer or carbon fiber reinforced polymer plates to facilitate the production.



Figure 4-93: 3D Printed Prototype of MOVE-II

The first usage of the prototype was the development of the cable harness. With an exact copy of the real hardware, the exact positions and lengths of the cables were relatively easy to determine. This process was faster than the traditional approach and allowed us to build the complete Engineering Model cable harness before having a single piece of EM hardware available. Also, it was possible to develop the integration procedure by assembling the prototype several times without harming operational hardware. We were able to refine our mechanical and electrical design and also to develop GSE by testing the assembly process. The decisions for design changes were facilitated by implementing all alternatives in the prototype and assessing them cost-effectively. Minor mechanical collisions due to design changes could be fed back to the circuit board design with a demonstrative object of study. We could proof the basic functionality (e.g., deployment of Flappanels) at all points in time using the prototype, although a detailed tolerance assessment was prevented by the manufacturing inaccuracies due to the rapid prototyping production processes. Due to the early integration of the prototype, GSE and specific tools could be defined ahead of time. Rapid prototyping methods were used to produce this equipment to ensure an agile development process.

Furthermore, several stands for the satellite have been developed to ensure a safe and reproducible integration of the satellite. The challenge was to consider different attachment points for holding mechanisms for different integration steps. Additional stands and test equipment were produced to allow fast and reproducible tests such as an automated deployment pin release mechanism for TV deployment tests. To test the deployment, we had to reset the Hold Down and Release Mechanism (HDRM) on a regular basis. This process is very complex and has the potential for damage on the solar cells. A reset tool was designed and produced by FDM to make it safer and reproducible. The satellite is held by a plastic frame and the Flappanels are pushed into the exact reset position by arms attached to this frame. That made it

possible to push down the ADM by hand without any further adjustment while the HDRM is open. We are able to reset the satellite within ten minutes for the next deployment test due to the shape memory alloy based design of the HDRM and the reset tool.

This method is just one example of multiple efforts to shift risk upfront in MOVE-II. The high-altitude balloon flights of both transceivers and the CDH subsystem is another example. Normally, engineering and prototype models of spacecraft are mainly used to detect problems associated with design, quality control, materials and operating procedures. Later, flight models are tested under simulated environments and the expected stress levels to detect workmanship problems and early failures, and assure and accepted level of confidence prior to launch, as Timmins noted already in 1966 [75]. For CubeSats, limited resources, tight schedule and less experience/heritage than in traditional missions are the reasons why we have to shift the risk upfront, and we will have to do that in most cases with very limited budget. The HDRM of MOVE-II, 2SMARD, is an example, where we tried to shift risk upfront and minimized the chance of DOA/infant mortality. Flying mechanical systems often means imposing an even greater risk to the system than with electrical or electronic systems, since they have a greater likelihood of causing catastrophic failure or loss of mission when they fail. Furthermore, most mechanical systems lack heritage as they are often first-time applications, do not allow repetitive testing, and suffer from long periods of storage [28]. 2SMARD, due to its design based on SMA, allows repetitive testing. The system on the FM was actuated more than 100 times before launch⁸¹. Also, as already pointed out, we tried to do early component and subsystem level environmental testing to reduce the risk of such issues arising when testing on system level. For 2SMARD, we had the mechanism both on a shaker, simulating launch vibrations, and actuating in different conditions in our TVAC, as soon as the first hardware model was completed.

Shifting the risk upfront is also necessary for the software development of CubeSats. As we have seen in earlier chapters, the significance of software for space missions, the complexity of software and the rate of software errors on space missions constantly increases. Software reliability cannot be predicted beforehand, it can only be measured, but it is important to understand that this process can also be started before the entire satellite is completed [21]. Thus, the software of MOVE-II was developed using an agile approach. Agile approaches are iterative processes in which the created artifacts evolve incrementally. Therefore, these approaches are well fitting for changing environments and have the advantage that the result of any iteration can be delivered to the customer. This gives the benefit of having a usable product early in the process and thus allows early testing and the delivery of a so-called minimum viable product at almost any time. Modifications and improvements can then be iteratively delivered in following versions and the previous versions can always act as a fallback solution if the newest version does not work as intended.

Software development for space applications is affected by historically grown structures and often conservative methods. Often, the associated uncertainties of space projects led to software being one of the most critical aspects of success and failure of current space missions. Yet, traditional processes offer only limited flexibility for the often-changing requirements, planned resources and schedule. This results in processes that are highly-time consuming and in many times also costly to implement. For educational CubeSats, with their fluctuating number of members, limited resources and tight schedule, these traditional methods would result in too little flexibility and too much management overhead. Thus, to shift risk upfront, and to get a stable version of the software for testing early on, we used an agile approach for the software of MOVE-II. Using a set of initial requirements, the minimum viable product, i.e., the first version of all the software components, was implemented within a few days. This first version included only the most important functionality and interfaces but was critical for early system level testing, and many problems in the interaction between subsystems and their software were detected and resolved due to this approach. Also, throughout the development process, it was always possible to roll back to the last stable version in

⁸¹ A long duration actuation test with EM hardware is still ongoing with more than 1,000 actuations and so far, no loss of function occurred.

case a new feature did not work as intended. For the incrementally change of software, automatic build pipelines, the version control system git and the web interface GitLab helped us to deploy new version of the software and test it on the system. Furthermore, a review process within the team helped to incorporate changes to our tested version only if more than one developer agreed to it, assuring high code quality and a broader knowledge within the team on the changes, known issues but also the overall progress. Combined with a Scrum approach, this enabled us to have new versions of the software every week and test these new features continuously on the satellite. The overall approach was not only implemented on the software of the satellite itself, but also on the software of the novel mission operations interface (see Figure 4-94). To test the complete chain from the ground to the satellite (missions interface – ground station – antenna – satellite), a first stable version of the mission operations interface had to be available early on and was subsequently improved, thus using also an agile approach.

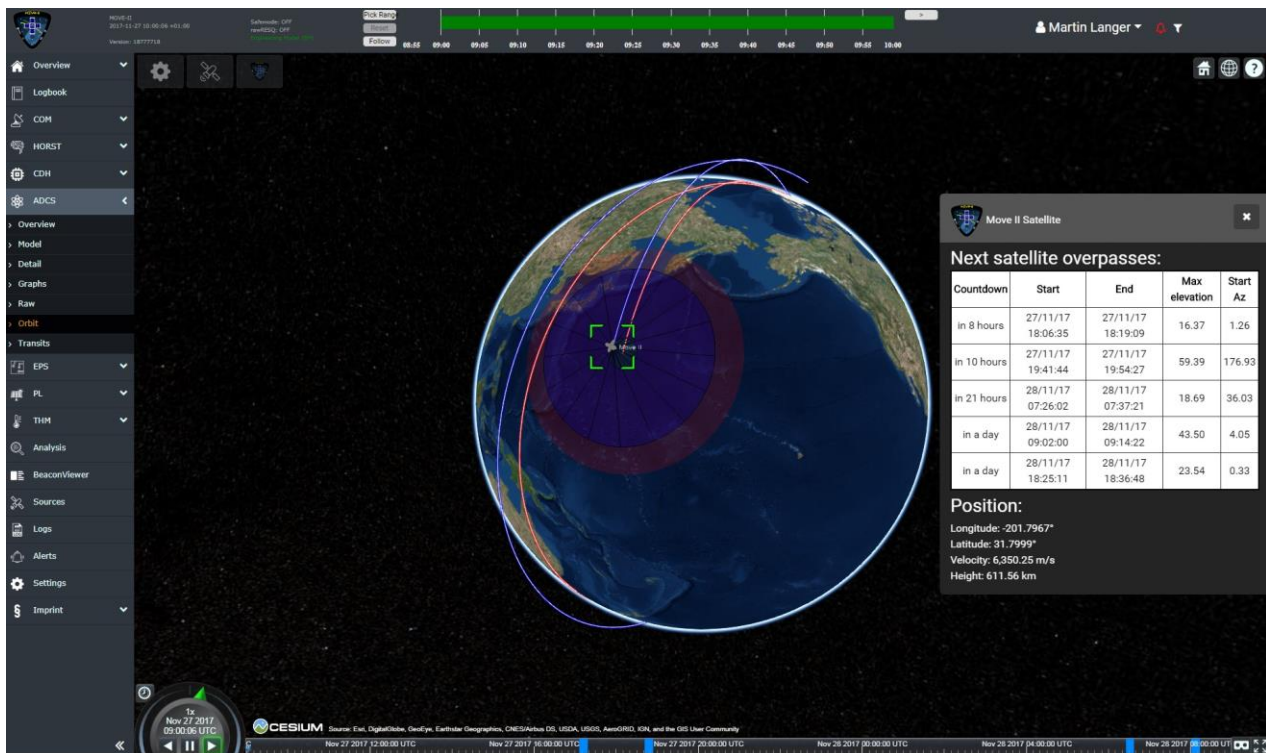


Figure 4-94: Missions operations interface of the MOVE-II mission.

A testing environment for subsystems does not have to be complex nor expensive, as the example of the low-cost ADCS testing environment of MOVE-II shows. To test the ADCS independently of all other subsystems, it needed a power supply and a data interface. At the beginning of the development we used several breakout boards to supplement these interfaces. The successor to this solution was a PC/104 compatible board, emulating the functionality of the CDH, EPS, and COM. The carrier board is depicted in Figure 4-95. The CDH and COM are substituted through a Beaglebone Black Wireless (BBBW). A BBBW is small enough to fit on a CubeSat-Kit sized PC/104 PCB without any alterations. The Beaglebones SPI, Inter-Integrated Circuit (I2C) and general-purpose input/output (GPIO) pins are directly connected to the PC/104 bus. Via the Wi-Fi connection, the developers can log in to the Beaglebone remotely. This enables them to issue commands and log sensor data even from their homes. Two serial connections allow logging of the debug outputs of the ADCS.

The emulated EPS on this board features an efficient 5 V, 2 A step-down converter and a 3.3 V, 500 mA converter. The converters are internally over-current and over-temperature protected. Both voltage lines are equipped with a polyfuse matched to the ADCS maximal consumption. The polyfuse allows our developers to create short-circuits while testing, without causing damage. Two power sensors continuously

measure voltage and current on the Beaglebone, enabling developers to directly assess the effectiveness of power saving techniques. Power switches allow the 3.3 V and 5 V lines to be turned on and off remotely. Two 9.6 Wh Lithium-Ion batteries allow up to 5 h of autonomous operation. This is needed for long-term tests (e.g., continuous detumbling tests) and thus long-term operation of the ADCS. A battery charger capable of charging 1.5 A of current is wired to the PC/104 external access charging line. It enables charging of the batteries through the MOVE-II common external debug interface.

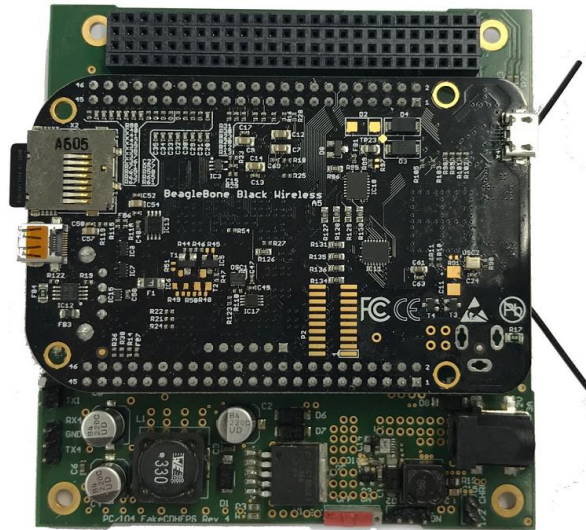


Figure 4-95: The PC/104 compatible carrier board for a Beaglebone Black Wireless. It also includes basic features of an EPS necessary for testing.

For validating the control algorithms, a real-time simulation containing the space environment as well as actuator and sensor models was connected to the Mainpanel of the ADCS. Together, they form a closed control loop that can simulate the ADCS in all mission phases. The ADCS control loop is shown in Figure 4-96. It includes the Mainpanel as the controller, the actuators on the Sidepanels, disturbance torques stemming from the parasitic dipole and the atmosphere, the space environment, spacecraft dynamics, and sensors on the Sidepanels. For simplicity, the Toppanel is referred to as a Sidepanel. Many CubeSat teams arrange a setup with a Helmholtz cage, a low friction bearing, such as an air bearing or a thin wire, and a sun simulator to verify the function of their ADC systems. This approach has the advantage of including all components of the ADCS in the test. On the other hand, the friction of the bearing limits the maneuverability in the test setup and greatly affects the dynamics of the satellite during testing. We used a Helmholtz cage to verify the B-dot detumbling algorithm. For the HiL testbed, the only dedicated hardware is the Mainpanel and auxiliary boards for interfacing between the simulation PC and the Mainpanel. The actuators and sensors are approximated with models resembling the worst-case noise and bias of their real-life counterparts.

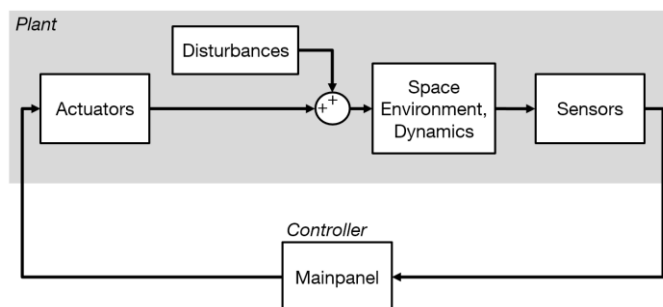


Figure 4-96: ADCS Control Loop.

The plant shown in Figure 4-96 is implemented as a real-time Simulink model, which also contains a Simulink implementation of the control algorithms to verify the firmware running on the Mainpanel. The simulation personal computer (PC) is connected over Ethernet to a device called Panel Emulator, which is equipped with a Beaglebone Black single-board computer and five microcontrollers resembling the Sidepanels. The Mainpanel is connected to the Panel Emulator in the same way as to the Sidepanels and runs in flight configuration. The stack of the Mainpanel, the Panel Emulator, and the ADCS testing board described in the previous section is shown in Figure 4-97. The simulation and the Mainpanel parameters can be configured with a MATLAB script that allows automated testing of different controller gains with varying environmental conditions. Thus, both the detumbling as well as the sun-pointing controller were verified with the HiL-Testbed.

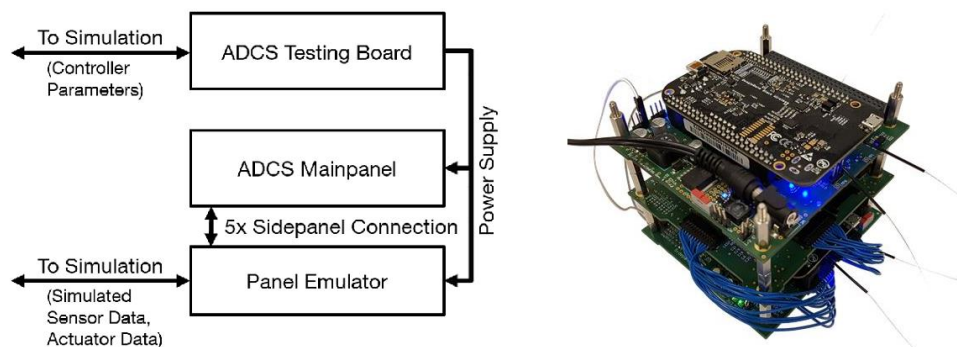


Figure 4-97: Mainpanel in the HiL-Testbed between the ADCS Testing Board and the Panel Emulator. Image Source: [277]

The detumbling controller uses the B-dot algorithm. With a worst-case initial velocity of 0.5 rad/s in every axis, the detumbling controller reduces the angular velocity to 0.01 rad/s in every axis within 162 minutes or 1.7 orbits. The detumbling behavior is shown in Figure 4-98. Reducing the velocity from 0.1 rad/s in every axis, which is a more realistic scenario after separation from the launcher, to 0.075 rad/s in every axis, from where the satellite switches to sun pointing mode, takes 12 minutes.

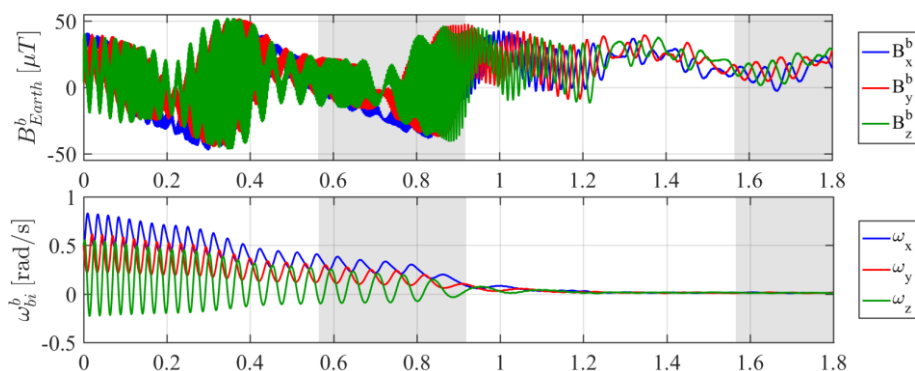


Figure 4-98: B-Dot Detumbling Controller Spinning Down from (0.5 0.5 0.5) rad/s to (0.01 0.01 0.01) rad/s. The Eclipse is marked in grey.

In sun-pointing mode, a state-feedback controller stabilizes the satellite in a spin around its z-axis at 0.1 rad/s and maneuvers the spinning satellite's top face towards the sun. Figure 4-99 shows four orbits with the sun pointing controller enabled. The pointing error is 21 degrees on average, which is primarily caused by the parasitic dipole that was estimated to be 0.02 Am² in the worst-case. Later measurements showed that the real parasitic dipole of MOVE-II has a value of 0.007 Am² to which the ADCS responds with a reduction of the pointing error by a factor of two. The lower graph of Figure 4-99 shows the energy charged into the battery. It considers the generated solar power, the power consumption of the ADCS (avg. 0.76 W),

the computer (avg. 0.4 W), the Electrical Power System (quiescent consumption of 0.6W plus conversion losses), and the UHF/VHF transceiver (avg. 1.5 W assuming 8 min contact on every orbit). After 10 minutes, the pointing error is reduced to 55 degrees and the solar panels generate enough power to start charging the batteries. The pointing error reduces the solar power by 8% compared to a perfectly sun pointed attitude, which is acceptable. The positive estimated power-budget in worst-case conditions builds confidence in the ability of MOVE-II to allow continuous operation of the basic subsystems and charge the batteries for short-term operation of the payload and the S-Band transceiver.

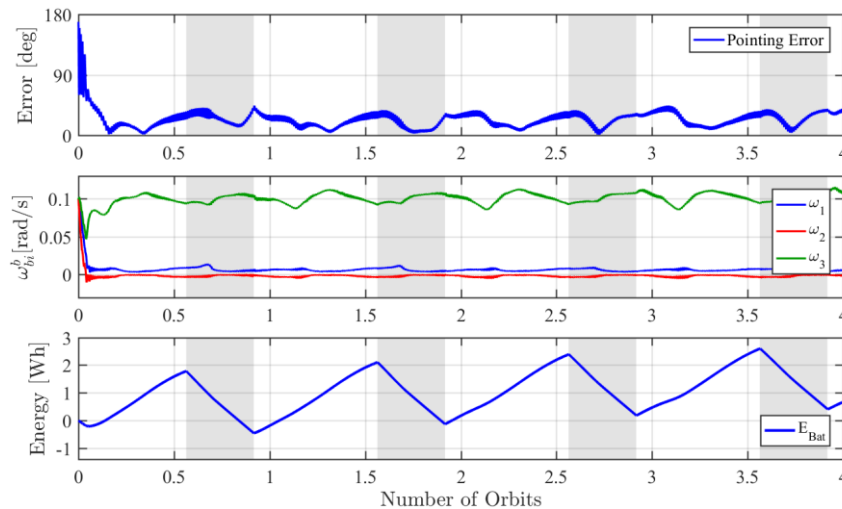


Figure 4-99: Sun-Pointing Controller adjusting spin and reducing the pointing error. The Eclipse is marked in grey.

Creating a reliable system is by far the greatest challenge for any CubeSat team. As the satellite can only be reliable if all critical systems work flawlessly together, system-level tests are absolutely mandatory. However, as noted before, an insufficient approach to system-level testing is seen as one of the main reasons for the low reliability of CubeSats [11]. In terrestrial applications, certification periods are common as the final process before the product is put on the market. These certification tests can range from weeks to several months, and often involve staff that has the greatest engineering knowledge and judgement of the product [289]. On the other hand, a growing number of software companies relies on public beta testing to achieve this step before commercial release [290]. For space projects, we already learned that end-to-end testing and the TLYF approach are two methods that have to be embraced for a reliable product [60]. As Trela & Maximoff noted [291], increasing/maximizing your test space coverage should be another goal to pursue in system-level testing. Finally, all test data produced have to be accessible and easily analyzable so that bugs can be solved, and the reliability of the system measured. For all three areas, we implemented CubeSat specific solutions that will be presented in the following. As in any other university based CubeSat project, we encountered challenges while doing so: time pressure, produced by the delay of the previous project phases; dependence on systems that are not developed far enough (in case of in-house developments); delayed delivery (several months) on COTS subsystems and lacking experience in test planning.

To address some of the challenges of system level testing, we decided to set up a remote-controllable testing environment for the satellite (see Figure 4-100). The command line interface was made accessible via remote tools as well as a logic analyzer and a multimeter, reading all logical signals and voltages on the bus as well as a camera recording all debug light-emitting diodes (LEDs). This enabled developers to test fast and efficient without having come to the facility and reduces the risk of hardware being damaged by testers. Apart from this, the following principles were applied: All encountered bugs are documented in a ticketing system, all quantitative data streams are visualized to find irregularities and patterns, the system is online 24/7, and the fear of breaking something is no sufficient excuse not to test. As O'Connor [134]

noted, a reliability demonstration test is never as effective in generating reliability improvement as a test planned, to onset as many failures as possible. Bearing that in mind, competitions to onset (and subsequently resolve) as many bugs as possible were made in MOVE-II⁸². Thus, through a “gamification” process, the system level testing time and heterogeneity of tests vectors were maximized in MOVE-II, involving as many testers as possible by 24h remote accessibility.

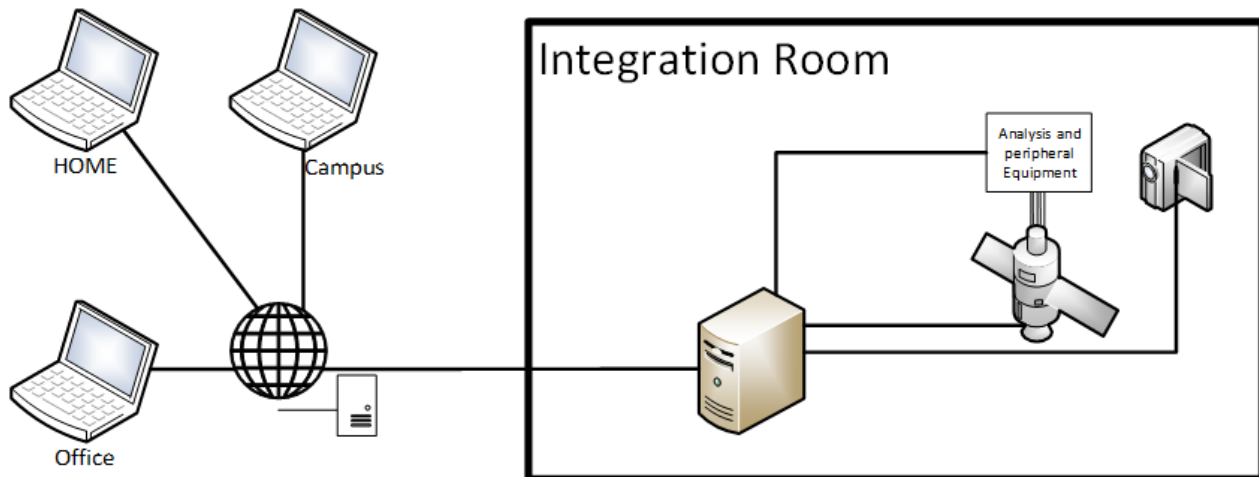


Figure 4-100: Remote testing setup of MOVE-II.

The mindset of “revealing as many bugs as possible” is especially important in our view and was applied on environmental and operational tests as well as through tests on critical malfunctions such as insufficient power for continuous operation. As McCurdy notes [14], failure is a normal by-product of any complex task, and is one of the primary ways by which engineers learn how to improve their designs. As we have seen in earlier chapters, advancements in parts design and manufacturing over the last decades led to very high part quality of COTS components [237]. Thus, part failure is one of the lesser reasons to worry about when building and testing a CubeSat. Increasing complexity and software caused an increased likelihood of latent design and workmanship defects being present in the system at late stages [37]. We have learned in earlier chapters that software failures are latent design errors, thus they cannot be prevented by redundancy or other traditional strategies of failure mitigation of spaceflight [59]. As said before, software reliability can only be achieved through testing. Although both software LOC and complexity is increasing, empirical studies show that not all combinations of settings have to be triggered in order to achieve exhaustive testing [292]⁸³. As noted by Frazier et al. [237], reliability efforts are best spent on known weak links, which in the case of university-built CubeSats is best done through system level testing, in order to uncover engineering flaws and other inconsistencies in the system.

In MOVE-II, we started system-level FlatSat testing in January 2017, one year before the planned launch date (and 18 months after the project started). All subsystems were connected by an external harness, giving access to all electronic connections. Through the FlatSat test-setup, which can be seen in Figure 4-101, many interface problems and software flaws were detected in the system. In March 2017, the EM was integrated completely for the first time, last modifications of the hardware were conducted and environmental tests (shaker, acceleration, thermal vacuum testing) were carried out. In addition, as the main challenge for any CubeSat is to overcome DOA and infant mortality, 24-hour tests were conducted regularly in our test-setup. The goal of these tests was robust testing of the first 24 hours of the satellite’s operational

⁸² Especially the “battle of the bugs” between the ADCS and CDH team was very fruitful for that, and will be discussed in Subsection 4.3.2.

⁸³ Kuhn reported of a system with 20 inputs, each of which can assume 10 possible values. Instead of requiring 1020 test cases, he showed that nearly all failure cases could be triggered by at most six erroneous parameters [292].

lifetime, including deployment of the solar panels, first contact, disabling of the Launch and Early Orbit phase (LEOP) sequence, and enabling all systems that are deactivated during LEOP. These tests proved to be very effective in showing up faults of subsystem interaction and software interaction. After those tests, the EM was left in continuous operation mode to be accessible for testers via the remote testing setup. In parallel to the EM tests, the FM was produced. Starting in July 2017, the FM was ready for testing. In addition to environmental tests and further sensor calibration and functional testing, tests under operational environment conditions were conducted in with the FM. Communication via debug interfaces was replaced by the satellite's UHF/VHF link. For these tests, access to the systems was later limited to a set of six overpasses per day during which software updates and tests were conducted. These tests especially helped to verify the usability of important system analysis tools, file transfer mechanisms and reliability of the complete space - ground communications chain including the missions operations interface. Benefiting from the two-model philosophy, both the EM and the FM of MOVE-II are tested in parallel since then. Corrections of bugs are first implemented on the EM to test the stability of the patch, and later applied on the FM. Another important method while working with two clones of the satellite was to use Quick Response (QR)-coding for all subsystems and parts of the satellites, in order to track test data and the history of any part of the system. A more detailed analysis the system-level tests is presented in the next subsection.

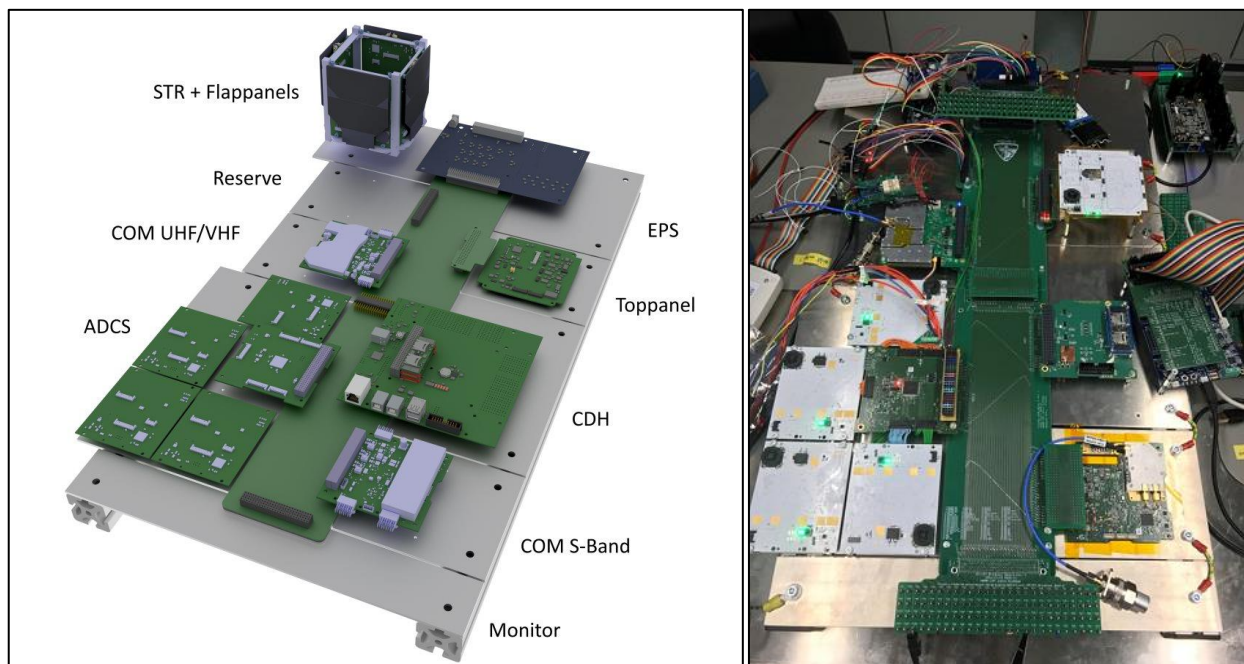


Figure 4-101: Rendering (left) and photo (right) of MOVE-II FlatSat.

Finally, while conducting system level tests with heterogeneous test vectors and uncovering as many bugs as possible is an important step in order to increase the reliability of CubeSats, the tracking and management of failures through a FRACAS must also be managed for that purpose. Through an interdisciplinary project we achieved to introduce a semi-automated FRACAS, called Elfriede, in MOVE-II. Reporting and correcting failures in an inefficient and unreliable way is an obstacle that led to problems in past space missions, independently of the resources involved. As Leveson [60] reported, limited communication channels, poor information flow, and diffusion of responsibility and authority were amongst the main reasons while the space missions she studied failed. In another paper [61], she added that several space losses were the direct result from flaws in the anomalies handling reporting systems. Both the Titan/Centaur and the Mars Climate Orbiter accidents suffered from missing communication, i.e., the evidence that a problem existed in the software was there before the loss occurred, but there was no communication channel established (or the existing channel was too ineffective or unused) to get the

information from the people who knew about it to the people who could understand it and to those who made the decisions [61]. Also Hall & Blay [98] noted that overcoming attritional knowledge loss while investigating errors is essential, and immediate access to information about the error is needed. In a university-based CubeSat project such as MOVE-II, those points are of particular importance. Knowledge about the system and the behavior is often spread across the team, and information of errors and bugs has to get from the students that are detecting it, to the students that worked on the system, and also pass by decisions authorities, in our case the project management. Furthermore, fluctuation of team members and the general academic schedule can lead to delayed solving of known bugs, and thus knowledge loss about what happened is a looming problem for any university team. Nieberding [69] showed a few characteristics of a FRACAS for general spaceflight purposes that would help to get the communication going between those concerned about a specific problem and those in a position to reconcile it. Such a system must be formal, visible and reliable, and should be simple to use with quick feedback. It has to be culturally valued and a real authority (in our case the project management) has to be plugged in to make decisions [69].

From the beginning of the development, MOVE-II relied on the project management software Redmine [293], in which amongst others, the planned developments and later the detected issues were tracked using a built-in tracking functionality. Figure 4-102 shows an example of a bug ticket issued in MOVE-II. Through the ticket, the bug is specifically assigned to one person of the team, who has then the responsibility to resolve it and assign it back to the person who issued it for review, test, and if possible, closing of the ticket. The project management reads the bug tickets on a regular schedule and checks what tickets are delayed, and subsequently asks the responsible person for the reason for delay. The ticketing system allows also the assignment of more than one person, which is sometimes needed due to the complexity of the occurring problem. In the description of the ticket, the information about the bug, and in what kind of test setup it occurred is depicted. Furthermore, attachments such as photos, measurement protocols and code snippets can be added to the ticket simply by uploading them to the online system. As Redmine is an online project management tool, it is easy accessible and sends updates automatically in case a ticket is updated or newly issued. Thus, independent from the location of the test and the testing crew (often bugs were detected from someone testing the satellite remotely), issues and problems can be reported quickly, are visible in project, can be tracked along the project life and cannot get lost, since each bug has its individual issue number. As depicted in Figure 4-102, the comment section is used to present the solution to the bug in order to allow the review of the implemented corrections. As already pointed out, the approach with git and GitLab for all software changes on the satellite, also visible in the comment section, allows the roll back to the last version in case the implemented bug fix does not work.

At the beginning of the testing phase, the issuing of bug tickets was done manually by the person who discovered the bug. However, as the project grew to about 100 active members in 2017, some issues arose from the parallel use of our main communication tool within the project, Slack [294], and the use of Redmine for bug tracking. The communication application Slack was introduced into the project to ease communication amongst the members of MOVE-II. The application can be used on mobile devices and different browsers and allows group and direct communication as well as the exchange of data. As the project grew, so did the traffic on Slack and up to May 2018, over 300,000 messages were sent over the MOVE-II Slack in less than 2 years. As some of those messages also described and discussed bugs, a manual follow-up documentation was always needed in case someone discussed an occurring bug that no one put into Redmine so far. Due to the missing integration between Slack and Redmine, this process involved several manual steps, which decreased the simple-to-use characteristic a FRACAS should have.

Bug #3546
[Edit](#) [Log time](#) [Watch](#) [Copy](#) [Delete](#)

Sidepanel X- and Y- BMX sensors not working after SLEEP mode

Added by [Florian Mauracher](#) 9 months ago. Updated 8 months ago.

Status: CLOSED Start date: 08/24/2017

Priority: Normal Due date:

Assignee: [Florian Mauracher](#) % Done:

Category: - Spent time: 6.00 h

Target version: -

« Previous | Next »

Description [Quote](#)

Observed voltages in Sleep mode on the sidepanels with disabled SPI bus on (X- and Y-)

Panel 1: 0.696V X-

Panel 2: 0.686V Y-

Panel 3: 1.068V X+

Panel 4: 1.062V Y+

Top: 3.3V

Subtasks [Add](#)

Related issues [Add](#)

Related to OMAC - Feature #3444 : D-Day	Closed	07/12/2017	08/31/2017	\$
---	--------	------------	------------	--------------------

Comments

History

Spent time

Updated by Florian Mauracher 9 months ago #1

https://gitlab.lrz.de/move-ii/adcs_software/merge_requests/163 [Quote](#) [Edit](#)

Updated by Florian Mauracher 8 months ago #4

https://gitlab.lrz.de/move-ii/adcs_software/merge_requests/166 [Quote](#) [Edit](#)

https://gitlab.lrz.de/move-ii/adcs_software/merge_requests/167

Updated by Florian Mauracher 8 months ago #5

The pullup on the CS line seems to be the source for the 0.7V residual voltage. [Quote](#) [Edit](#)

After PR !167 the voltage on the X- on Y- panels is 0V when powered off.

[Edit](#) [Log time](#) [Watch](#) [Copy](#) [Delete](#)

Also available in: [Atom](#) / [PDF](#)

Figure 4-102: Example of a bug-ticket of the MOVE-II bug tracking system within the online project management software Redmine. Status, Assignee, Description, Comments & Updates, Priority and Starting Date are amongst the most important fields to be filled out.

This problem led to several bugs being not tracked, thus not removed, and emerging again later in time. To solve this, an interdisciplinary project [278], supervised by the author of this thesis built an integration of our issue tracker into our Slack communication environment, called Elfriede. Figure 4-103 shows the general workflow of Elfriede. If a bug/issue/error message is sent on Slack, the simple addition of a bug emoji to the message triggers Elfriede to automatically create a new Redmine bug ticket and transfers the message to the ticket in Redmine. The successful completion of this process is then shown in Slack again, using a checkmark emoji (or a no-entry emoji in case Elfriede was not successful). The author of the message is automatically assigned as responsible person, as it was in the old process, and can then decide whom to assign to the ticket. Also, new messages written in Slack within the thread of the bug message are automatically added to the Redmine bug ticket as comments by Elfriede. In Figure 4-104, the typical appearance of Elfriede in Slack and the creation of the Redmine issue ticket is shown.

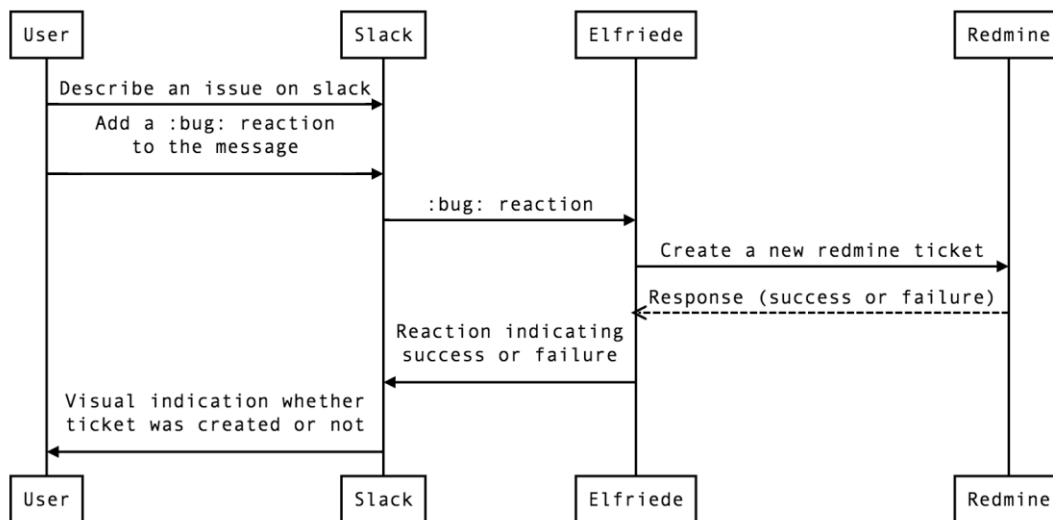


Figure 4-103: Workflow of the automated bug tracking in MOVE-II. Image Source: [278].

Overall, this combined efforts to detect, track and subsequently solve bugs was an important element of the system level tests of MOVE-II and the tracking of reliability growth throughout those tests, which will be discussed in the following subsection.

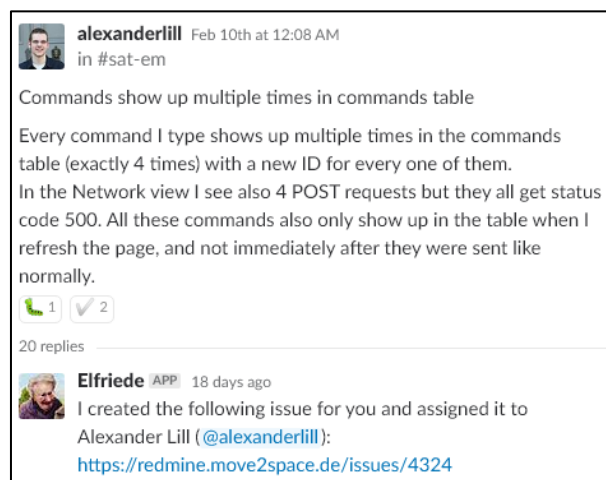


Figure 4-104: Typical appearance of the automated bug handling application “Elfriede” in Slack. Image Source: [278].

4.3.2 Assessing the Reliability of MOVE-II

This subsection is partially based on a conference paper [275] by the author of this thesis and the Master's Thesis of Florian Schummer [262], which was supervised by the author of this thesis.

Based on the overall test environment described in the last subsection, we will focus on the results of these system level tests in this subsection, and report on reliability growth models of the satellite. As already pointed out, system level tests were started in early February 2017 on the MOVE-II FlatSat. At this time, all subsystems were connected by an external harness, giving access to all electronic connections. It was then operated for the first time as a complete satellite. Before that, mock-ups and low-cost replacements were used while developing the subsystems to minimize surprises when connecting the final boards to each other for the first time. Starting at that point of time, all failures were tracked using the FRACAS (and resolved, if possible).

Despite the efforts to keep surprises and communication problems between the subsystems at a minimum, almost 100 problems (hard- and software) were revealed within the FlatSat phase. In March 2017, at around day 40 of the system level tests, the EM was integrated completely for the first time. Functional tests and 24h-tests were mainly conducted in the beginning, followed by environmental tests in simulated space environment. The EM test phase stretched out until late June 2017, when all qualification tests for the launch provider were completed. This included not only tests in thermal-vacuum, but also tests of the system under launch loads (vibration tests on a shaker, acceleration tests on a centrifuge). Late into the EM testing, the production of the FM was started. The first integration and subsequent functional testing of the FM started in mid-July 2017 (day 163). Until then, the tests on the EM and correction of failures continued. Similar to the EM, the FM underwent environmental tests and further sensor calibration and functional testing after July 2017 and passed the acceptance tests for the launch (thermal-vacuum, vibrational loads) in August 2017. Starting in August, the complete communication chain (missions interface – ground station – antenna – satellite) was used regularly for tests. This increased the overall number of detected errors again, since the communication now had to work not only directly with the satellite but also over the mission interface and the RF-link. Access to the system was limited to a set of six overpasses per day, during which software updates and tests were conducted, testing the system in the best possible TLYF manner. This especially helped to verify the usability of important system analysis tools, file transfer mechanisms and reliability of the space to ground and ground to space communications chain.

Beginning in late September 2017, the EM and the FM were tested in parallel, relying fully on the operations interface, which was updated from time to time to resolve bugs. Issues on the satellite were also corrected during that time, using again a TLYF approach: a correction for a revealed bug was first tested on the EM for stability, and only after a few days of error-free operation deployed on the FM, using the RF-link and mission's operations tools only. Overall, the recording of data for this thesis stopped on 12/23/2017, but the tests of the EM and FM will continue until delivery for launch. This short description of the system level tests conducted on MOVE-II shows necessity of both, environmental tests and system level functional tests. Without one of the two groups of tests, many errors on MOVE-II would have remained undetected. Figure 4-105 shows the cumulative number of errors detected in the space segment (EM and FM) of MOVE-II. Thus, issues arising from the use of the mission operations interface, GSE or ground station (GS) are not shown in here. The chart is segmented in four areas, which correspond with the already presented phases of system level testing⁸⁴. A total number of 432 errors were detected in the space segment within the observation window of 319 days. A saturation of the overall curve, but also regions of steeper increase in cumulative error number can be noted. The entry of the curve is relatively flat, which can be explained by multiple issues at the beginning of the FlatSat tests that prevented the full scale of system level tests to be conducted.

⁸⁴ I – FlatSat; II – EM; III – FM; IV – EM & FM.

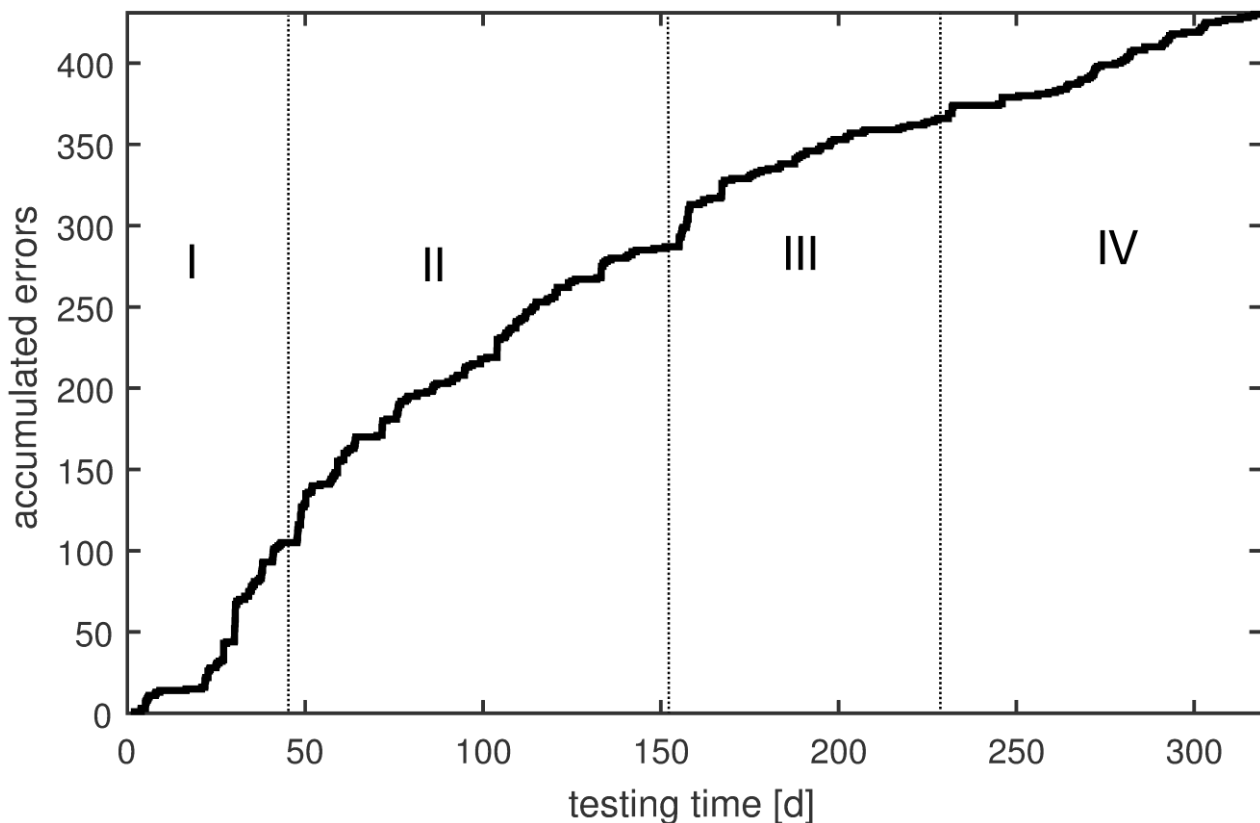


Figure 4-105: Cumulative number of errors over time on the FM and EM (the space segment) of MOVE-II. It depicts the region where system level tests were carried out on the FlatSat, II on the EM, III on the FM, and IV on both, the EM and FM in parallel.

In general, any anomaly that occurred within the observation window was recorded and put into the database. However, for this kind of analysis, the database had to be censored manually, since multiple errors had one common origin and some errors were reported more than once due to the time from first reporting to correction of error. As described by Ohba [179], this was handled by counting only the origin of the fault, and not the (sometimes multiple) outcomes, if possible. Also, some of the reports described not only one but multiple errors, and some of the anomalies seen resulted not from issues with the hard- or software but from mishandling or human errors. Based on the analysis of the error and the subsequent solution, we associated all of them to specific subsystems.

Taking a closer look at specific regions of the curve, the steep increases of cumulative errors around day 30 and 50 (marked (a) and (b) with green circles in Figure 4-106) can be explained by the first conduction of the 24h-tests. Multiple errors occurred within these tests that were related to the first day in life of the satellite and all errors were subsequently resolved. Thus, later 24h-tests cannot be recognized as well as the first two ones. Another increase in cumulative errors can be observed between day 100 and day 125 (marked (c) and with a grey square in Figure 4-106). This increase was due to a challenge within the team to find as many errors as possible in the system. This “gamification” helped to boost the number of people interacting with the satellite, and thus increased both the overall number of tests conducted but also the diversity of the test vectors applied.

Figure 4-107 shows another, later implementation of this gamification process. A notification window within the Slack environment informs every day publicly which person of the team issued the most commands within the last 24 hours. The fourth region marked in Figure 4-106 (blue circle and (d)) shows another increase in cumulative errors around day 150. This was caused by the project management team issuing a deadline due to the then upcoming FM integration and testing. Thus, as before, the number of people interacting with the satellite significantly increased in that period. Overall, a relatively stable number of

persons tested the satellites within the whole observation window. Of course, over the whole project the share of workforce spent on error correction decreased, while the share of testing increased, but on average the testing time per day stayed relatively constant.

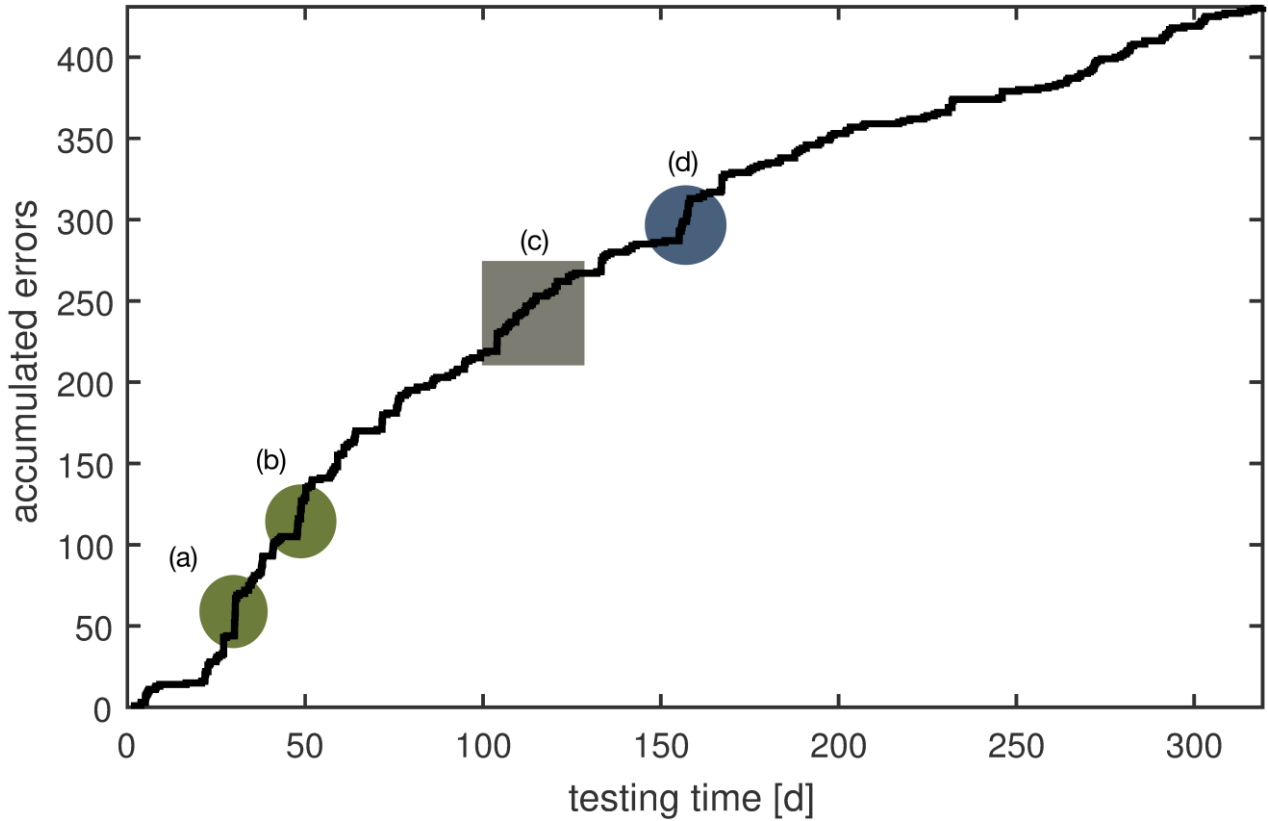


Figure 4-106: Analysis of specific regions of the number of cumulative errors over time of the MOVE-II space segment. Green marks areas in which 24h-tests were conducted, grey a region in which a bug-finding challenge took place, and blue a time of increased cumulative failures due to an approaching deadline set by the project management.

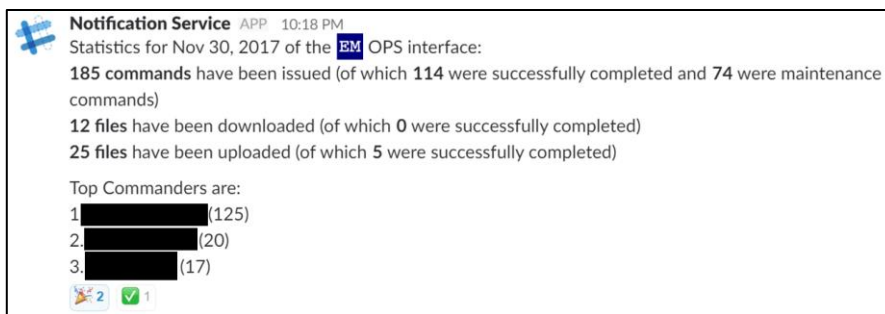


Figure 4-107: Notification Service of the MOVE-II Slack interface as an example of gamification of the testing process of MOVE-II to increase number of testers interacting with the satellite and diversity of test vectors.

Looking at the origin of the detected errors on the space segment, more than 82% stem from software on the satellite, leaving only slightly more than 17% as hardware-related (see Figure 4-108). This result is in line with data of larger satellites we have seen before (e.g. Brunner et al. [206]), although the fractions were slightly different for larger satellites. Thus, software plays also a significant role for the reliability of our CubeSat.

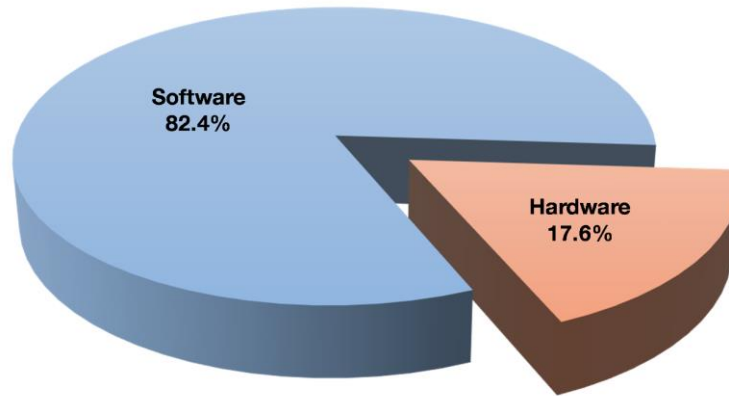


Figure 4-108: Origin of all bugs detected in the space segment of MOVE-II (432 bugs)

Figure 4-109 depicts the cumulative number of software errors of the space segment of MOVE-II over time. The overall shape of the curve is similar to the total errors on the space segment with the aforementioned regions of enhanced test effort clearly visible. Although a slight saturation of the curve can be seen, there was a good statistical chance of possible software errors still left in the system at the time of observation end. A total number of 356 anomalies related to software were found until day 319.

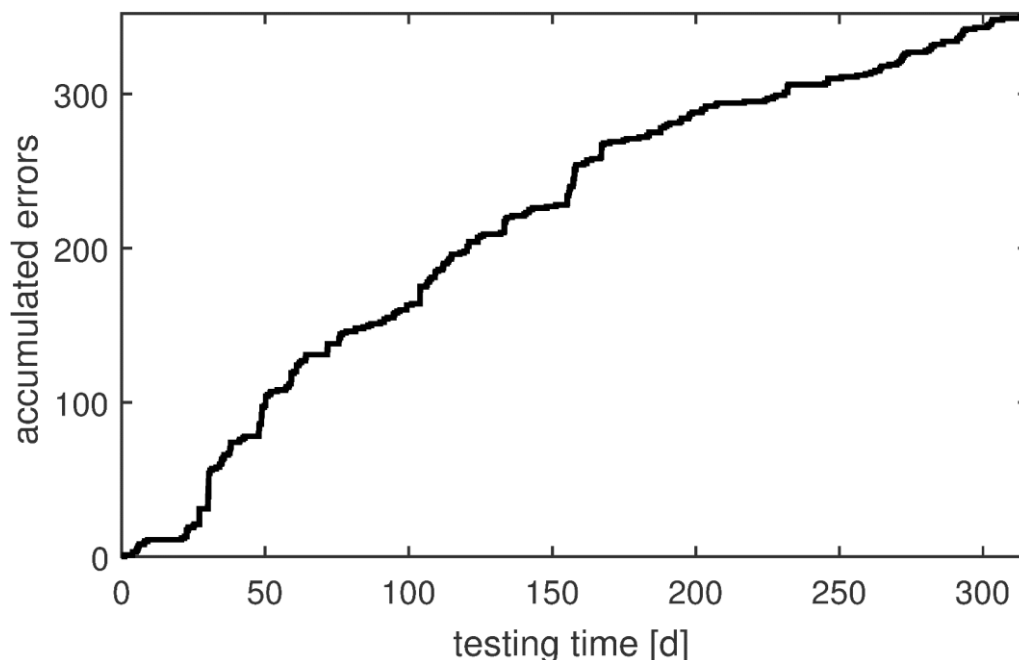


Figure 4-109: Software errors detected on the space segment of MOVE-II over time.

A clearer saturation can be seen in the hardware-related bugs, shown in Figure 4-110. A total number of 76 bugs related to the hardware of the space segment were found until the observation window ended. While looking at Figure 4-110, it shall be noted that the last error related to the FM-hardware occurred on day 220. As all data collected on the space segment comprise of both, issues on the EM as well as on the FM,

hardware errors on the EM caused a slight increase in the curve late in the observation window. This also includes evidence of wear-out on the anodic coating of the Sliders of the EM's HDRM after multiple hundreds of actuations for testing. Secondly, this saturation curve also incorporates errors emerging from environmental tests, not only functional testing.

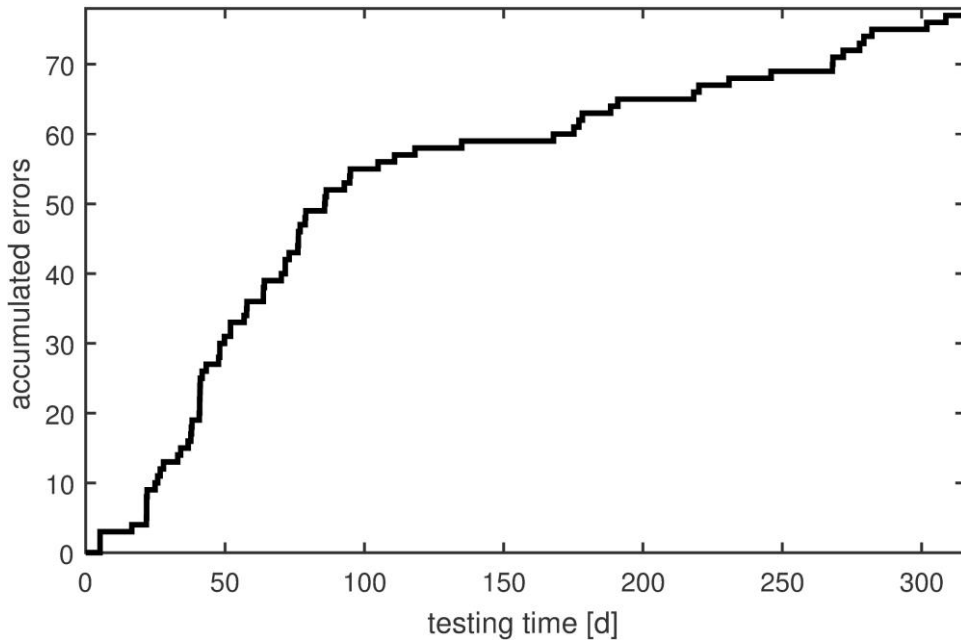


Figure 4-110: Hardware errors detected on the space segment of MOVE-II over time.

When looking at the number of overall detected bugs in the project (see Figure 4-111), it can be noted that no saturation occurs at the end of the observation window, and although the function is decreasing at around 200 days, the cumulative number of errors started increasing again at day 250.

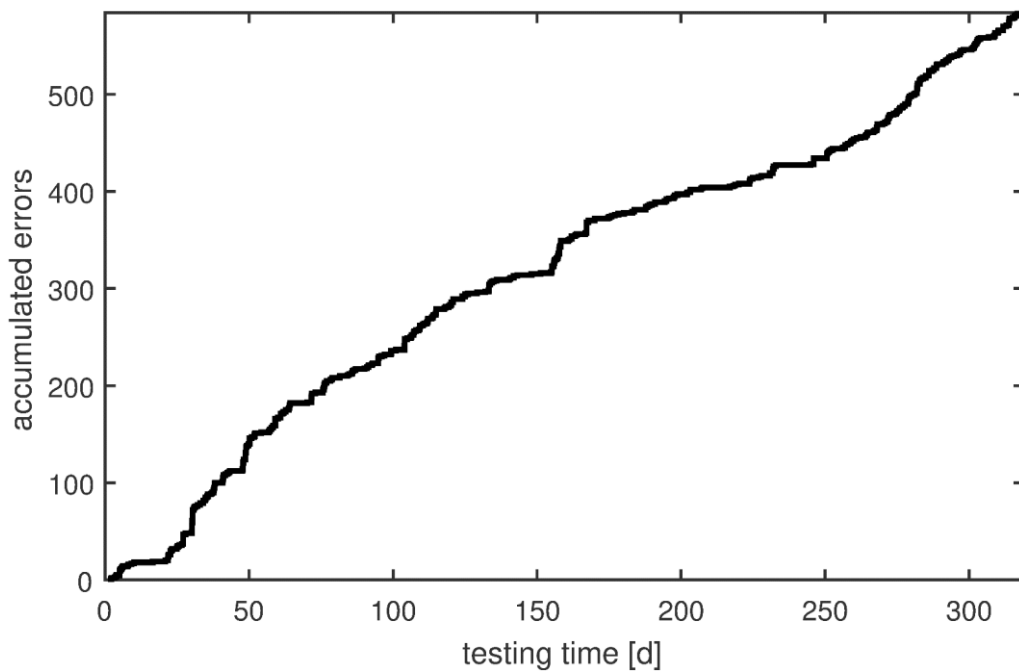


Figure 4-111: All errors detected in the MOVE-II system (space segment, ground station, missions operations interface, GSE)

The main reason for this behavior is the increased utilization of the mission operations interface after day 250 on two satellites in parallel. As already pointed out, the interface was developed from scratch and thus also suffered, as any new developed system, from errors that emerged when testing it in the complete communication chain. It should be noted though, that little to no of these anomalies would have caused a loss of the satellite in space. It would have ended up difficult though, to determine if an issue on the ground or the space segment prevented certain actions on the satellite to be executed. In addition to the errors in the mission operations interface, errors stemming from the GS and the GSE were also added in the MOVE-II system level statistics. Thus, it is rather a cumulative error number of the system of systems than of a single system. A total number of 583 bugs were detected until the end of the observation window, the majority again stemming from software rather than hardware (see Figure 4-112).

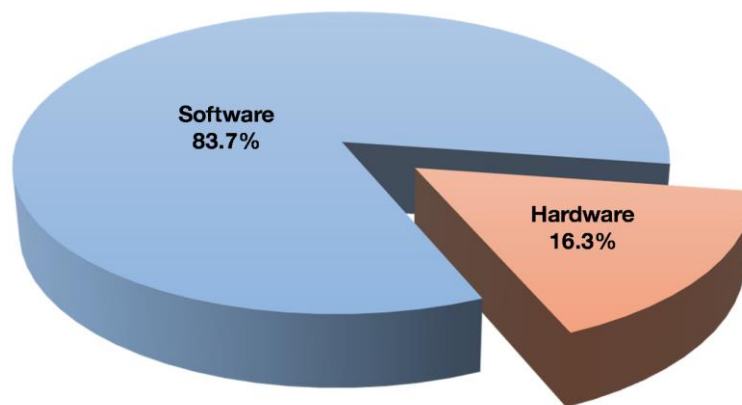


Figure 4-112: Origin of all bugs detected in the MOVE-II project (583 bugs).

Similar to the mission operations interface, an increase of bugs after day 250 can be observed for the ground station of MOVE-II for mainly the same reason as stated above. The accumulated errors of both parts of the MOVE-II system can be seen in Figure 4-113. The cumulative errors of the mission operations interface (OPS) (Figure 4-113 left) shows the aforementioned increase after day 250. Almost no tests with OPS occurred until day 100, and this can also be seen in the graph, as no errors were detected until then. For the GS a similar pattern can be noted. After an early setup on day 50, the GS was increasingly better incorporated in the system level tests over time. In combination with the heavy usage of OPS this led also to an increase of accumulated errors, not a saturation, after day 250.

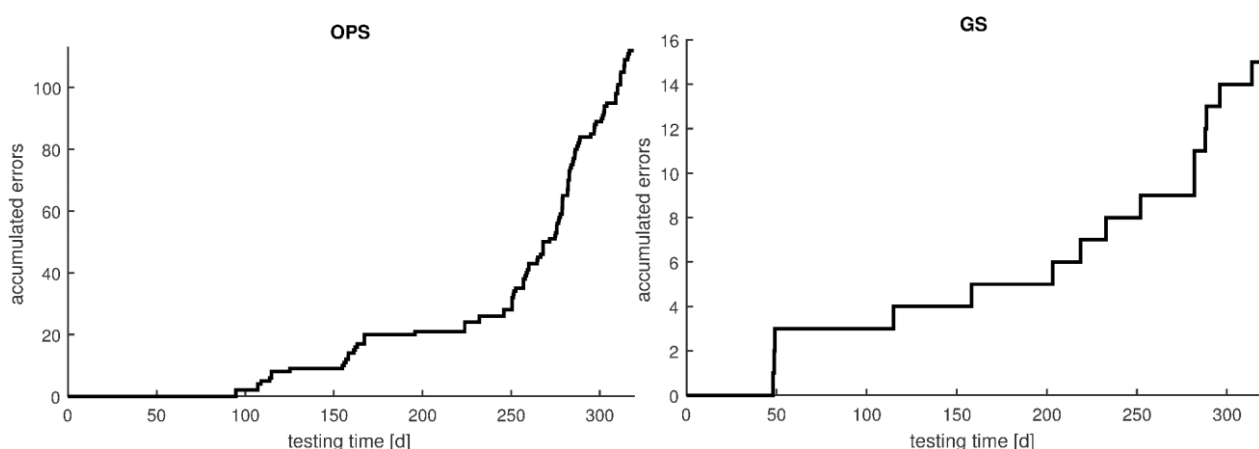


Figure 4-113: Cumulative errors of the mission operations interface (left) and the ground station (right) of MOVE-II.

Both systems are examples of why the system level tests of MOVE-II haven't stopped at the time of this writing (May 2018). Remaining bugs in the GS and OPS, although they can be resolved after launch, can cause major delays within the CubeSat's mission, which is unacceptable in our case (and presumably in most other missions) due to the already short mission time of most CubeSats. Also, as already pointed out, figuring out where a specific problem is originating from is much harder as soon as the satellite is up in space.

The two largest contributors to the cumulative number of errors on the space segment are the on-board computer including its operating system and software (summarized as CDH in Figure 4-114 left) and the ADCS of the satellite (shown in Figure 4-114 right). The first errors of the CDH were detected around day 25. Bugs in other subsystems prevented the occurrence of errors on the CDH. This masking is further discussed in Chapter 5. For CDH and ADCS combined, more than 100 errors have been uncovered during the system-level tests. This is followed by COM and the EPS, as depicted in Figure 4-115. COM shows also a delayed start of the saturation curve, while errors in the EPS increase right from the beginning, but flat out shortly after that and start increasing again at around 30 days. Both can be explained by errors in the EPS subsystem that occurred early and had to be resolved in order to test the complete functionality of the satellite. All errors are comprised of both hard- and software faults, with the latter being the more dominant source of errors for the four subsystems depicted here.

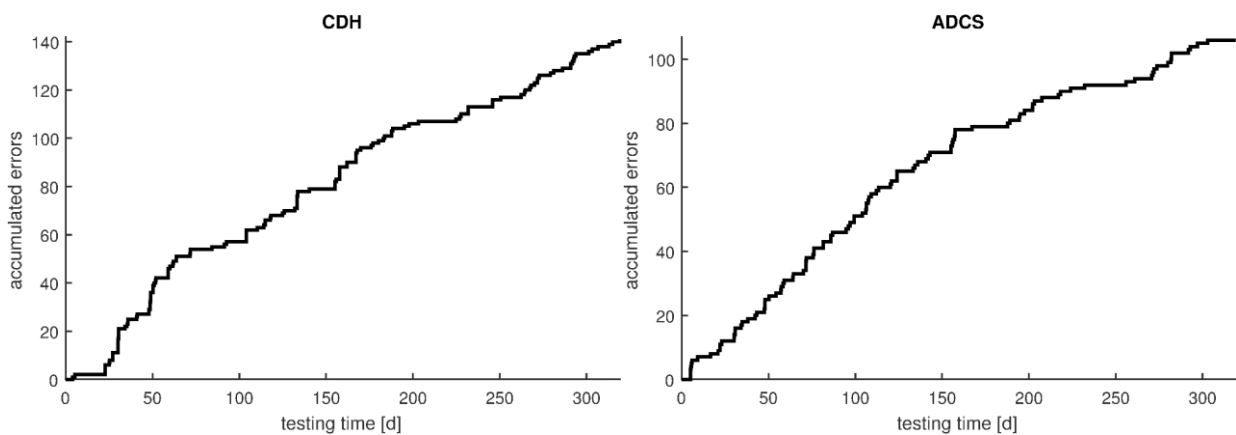


Figure 4-114: Cumulative errors of the CDH subsystem (left) and the ADCS subsystem (right) of MOVE-II.

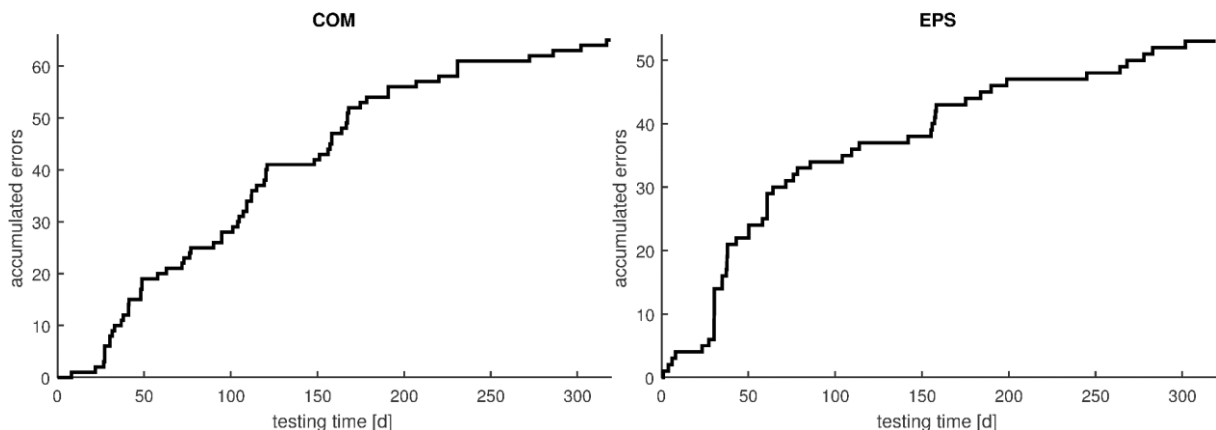


Figure 4-115: Cumulative errors of the COM subsystem (left) and the EPS subsystem (right) of MOVE-II.

The other subsystems of the satellite are shown in the following. In Figure 4-116 the accumulated errors in the GPS subsystem (left) and the PL subsystem (right) are presented. Both subsystems have a staggered entry but show an overall satisfying saturation. Thereby, the late errors in PL are software errors, which decrease the performance of the measurement, but do not impact the basic functionality of the subsystem.

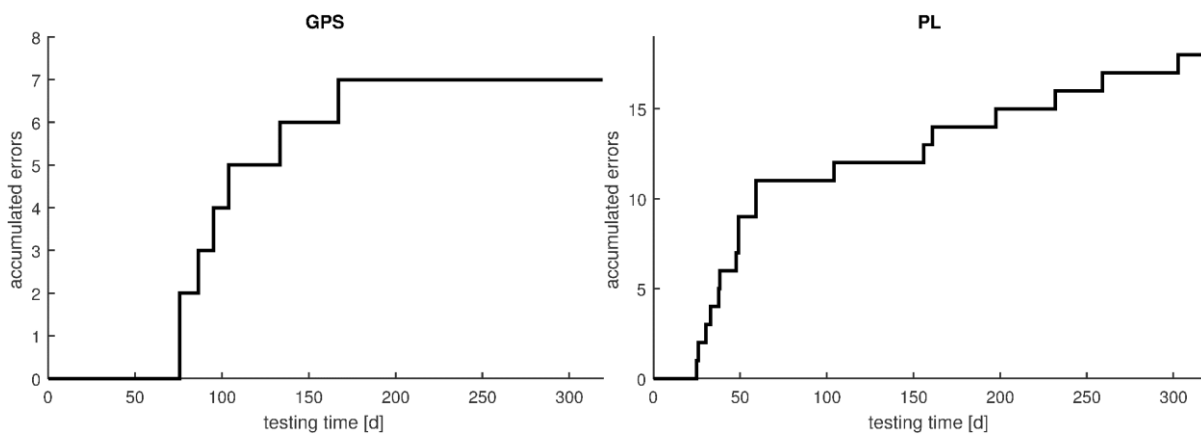


Figure 4-116: Cumulative errors of the GPS subsystem (left) and the PL subsystem (right) of MOVE-II.

As the last subsystems of the satellite, STR and the thermal system (THM) are presented in Figure 4-117. Thereby, THM captures issues with the temperature measurement on MOVE-II. Thermal problems on specific subsystems are considered as issues within the subsystem, as MOVE-II (and many other CubeSats) does not have an active thermal management system. For STR, a quick saturation can be seen after day 100. Mostly, issues in STR are hardware-related and normally do not mask other problems. Thus, when resolved and tested, new problems occurred seldom in this subsystem. One late error of STR can be noted: it is the aforementioned wear of the sliders of the EM's HDRM.

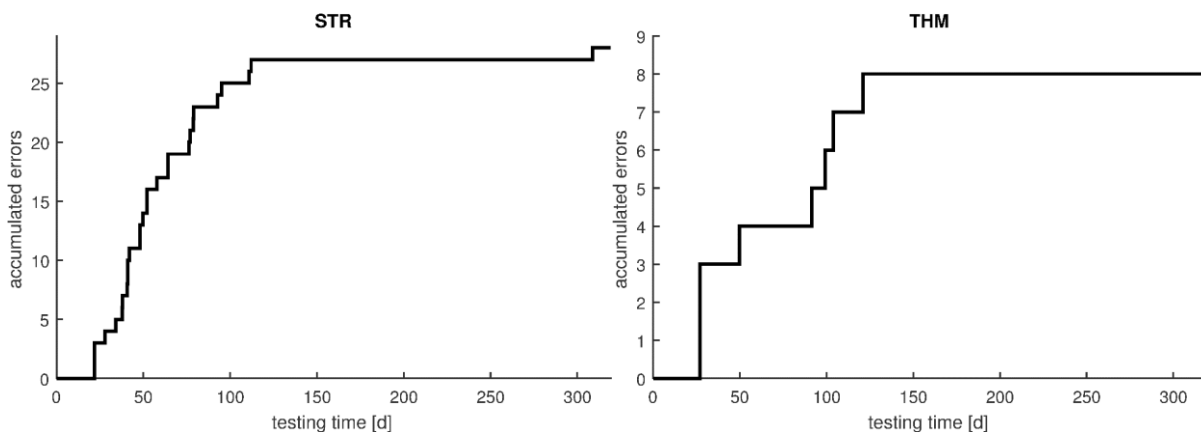


Figure 4-117: Cumulative errors of the STR subsystem (left) and the THM subsystem (right) of MOVE-II.

Finally, Figure 4-118 depicts the accumulated errors in the GSE of MOVE-II that occurred during system level testing. Sometimes, issues arising from the GSE stopped ongoing tests until resolved, so this is also an important indicator that not all errors in a CubeSat program are necessarily originating from the satellite itself. Also, it can be seen that errors in the GSE happened right from the start of system level testing, thus the interaction of the satellite with the GSE and possible errors originating from it should not be taken lightly in the beginning.

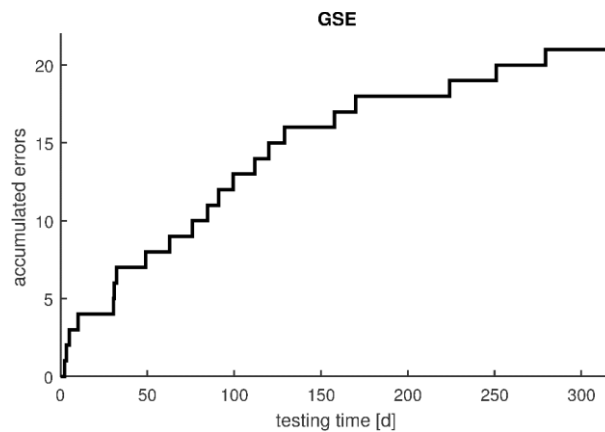


Figure 4-118: Cumulative errors of the GSE of MOVE-II.

The two-model philosophy implemented in MOVE-II turned out to be an important piece of the puzzle while searching for functional errors. As shown in Figure 4-119, more than 44% of all bugs were found on the EM or while using the EM. This can be explained partly by the more rigorous testing that can be put into place when using hardware that does not necessarily have to fly into space. On the other hand, it must also be considered that most EM testing happened before the FM was assembled, so many issues were just detected on the EM because it was earlier available for testing. Secondly, the two-model philosophy allows system-level tests of EM and FM in parallel, from which MOVE-II has also benefited. More than 27% of all errors were found while testing both models in parallel, thus doubling the testing time of a single-model approach⁸⁵.

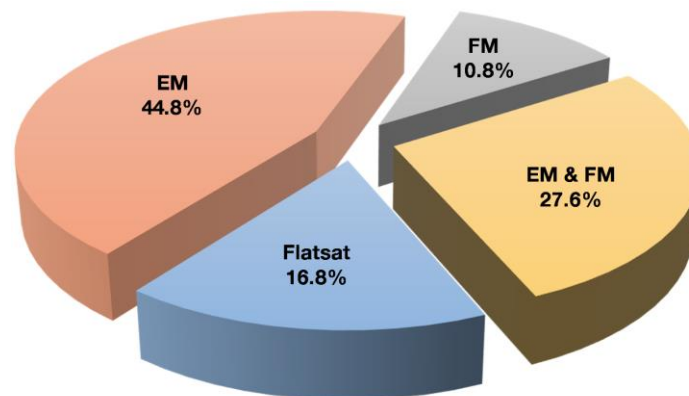


Figure 4-119: Fraction of errors occurring on different systems of the MOVE-II project (n = 583). Note that system level tests were first put in place on the FlatSat, followed by the EM, the FM and later both the EM and FM in parallel (see Figure 4-119).

As a last step of data analysis of the accumulated errors, we filtered all errors regarding their ability to cause a critical failure in space. 113 critical failures that would have prevented the satellite from working in space were identified. Although this filtering seems a rather subjective approach, it is necessary since many of the errors occurring in later stages of the MOVE-II system level testing were either limited to the mission operations interface or causing only performance issues, not critical failures. To decide whether to continue testing or not, or to decide whether to declare the satellite's readiness for launch or not, it is important to record and analyze both, the cumulative number of errors of the overall space segment but also the cumulative number of show-stopping failures over time. The filtering for critical failures should be done at least by the project management and the lead systems engineer, ideally accompanied by members of the respective subsystems. Figure 4-120 shows the cumulative number of critical failures in our system over

⁸⁵ Different response of the EM to the FM when using the same commands also helped for that.

time. In cases we were not sure if a particular error could cause our mission to end, we declared it as critical to be on the safe side. A clear saturation as well as a staggered entry can be observed for the function.

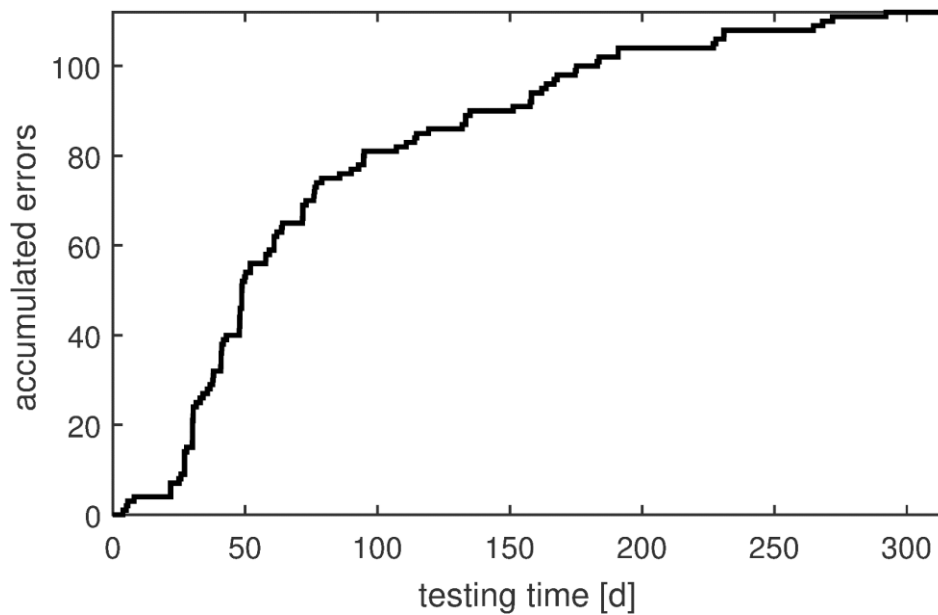


Figure 4-120: Cumulative number of critical failures of MOVE-II over testing time. We defined critical failures as failures that could potentially stop the mission.

When filtering only the critical failures, the ratio of hardware related bugs increases a little bit with respect to the ratios seen before. Figure 4-121 shows that more than a third of all critical bugs were hardware related. Still the majority was found in the software of the satellite.

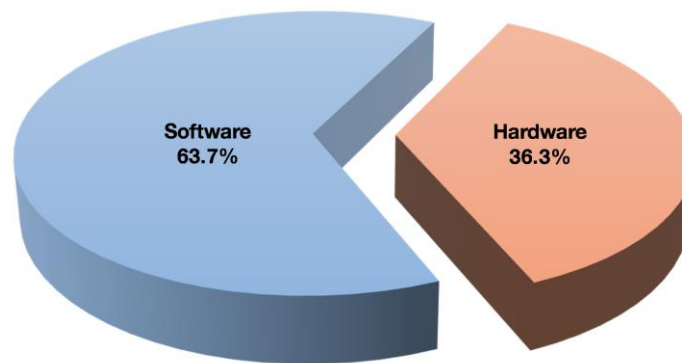


Figure 4-121: Origin of critical failures occurring in the MOVE-II project (n = 113).

In the following, we will present the reliability growth models used to analyze the errors on the space segment of MOVE-II as well as the critical failures identified in the system. The models, already presented in Subsection 2.2.2, will be introduced quickly and results of the analyses shown afterwards. The main reasons for using reliability growth models in MOVE-II were that we considered them helpful for the decision when to end system level testing and to deliver reasonable estimations of how many errors could be left in the system (both overall and critical errors). As any statistical model, the growth models cannot prove that the satellite will work in space, but they can show that the chance for early failures that originate from engineering/design/workmanship are reduced to an acceptable rate⁸⁶.

⁸⁶ The number of possible remaining bugs, both overall and critical, is given by the estimation. Thus, a value for this has to be agreed on by the project management of the mission, and traded off against schedule and resources. We will present the values accepted for MOVE-II, but other missions might have a completely different approach to risk, and thus can accept a different number of errors left in the system.

As the majority of overall errors and critical errors stemmed from software, five different software reliability growth models were used on the collected data. We used the basic exponential model of Goel & Okumoto [178], the delayed S-shaped model and the inflection S-shaped model from Ohba [179], the model with easy- and difficult-to-detect errors of Yamada & Osaki [180] and a modified exponential model with variable starting date, which we self-developed. All models are based on a NHPP, as earlier explained. The Poisson Distribution is suitable for data in which the numbers of times an event occurs are known but the number of times it does not occur are not known [36]. The used software reliability growth models basically need only the time when the error occurred as an input, which was provided by the FRACAS of MOVE-II. According to [180], a software reliability growth model based on a NHPP can be described by:

$$Pr \{N(t) = n\} = \frac{\{H(t)\}^n}{n!} \cdot \exp[-H(t)] \quad (47)$$

and:

$$t \geq 0 \text{ and } n = \{0, 1, 2, \dots\} \quad (48)$$

with n being the number of errors detected up to time t , $H(t)$ the expected value of $N(t)$ detected up to a time t , and t being a time interval larger than 0. Important for our purposes is that the total amount of errors in the system a (also sometimes called $H(t = \infty)$) can be estimated. All models were analyzed on their robustness of their estimation of the total amount of errors over different points in time. Also, the error detection rate per undetected error $d(t)$ can be modified in some models, as we will see later. First, we will start our analysis of reliability growth with all errors found in the space segment of MOVE-II, and continue later with the reduced dataset on the critical errors.

As described in Subsection 2.2.2, the basic exponential model by Goel & Okumoto [178] is describing $H(t)$ with an exponential function and only two parameters, the total amount of errors and the error detection rate per undetected error, which is a constant value. Figure 4-122 shows the basic exponential fit on the collected overall errors of the space segment over time. Although deviations in the first 30 days can be noted, the overall fit seems sufficient for our purposes. A total number of 507.1 errors is estimated by the model in the system ($t \rightarrow \infty$), which are 75 more than detected so far until the end of the observation window. The 95% confidence interval of this estimation is 489.9 errors and 515.3 errors, meaning that the remaining number of errors could be between 58 and 83 (rounded values). The error detection rate per undetected error is 0.0057 per day, with a 95% confidence interval of 0.0056 and 0.0059 per day. The overall goodness-of-fit is $R^2 = 0.9926$. At the time of observation, the model estimates 425.8 errors, with a 95% confidence interval of 404.6 errors and 447 errors.

To assess the stability of the prediction, the model was re-run with restricted statistical data from earlier points in time, namely 100 days, 40 days and 10 days before the end of the observation window. The results of this analysis can be seen in Figure 4-123. At day 219, the model estimated a total number of 561 errors in the system (rounded value), which is 54 errors more than estimated at the end of the observation window. At $t = 279$ days the estimation gets close to 500 errors and does not deviate much after that point in time⁸⁷. Thus, the basic exponential model seems to overestimate the remaining errors in our case at earlier points in time, which is considered better than underestimating it, but nevertheless a problem for trustworthy estimation of remaining errors. Figure 4-124 shows the four different estimations versus the underlying error data of the space segment.

⁸⁷ $t = 279$ days: 500.7 errors; $t = 309$ days: 505.1 errors.

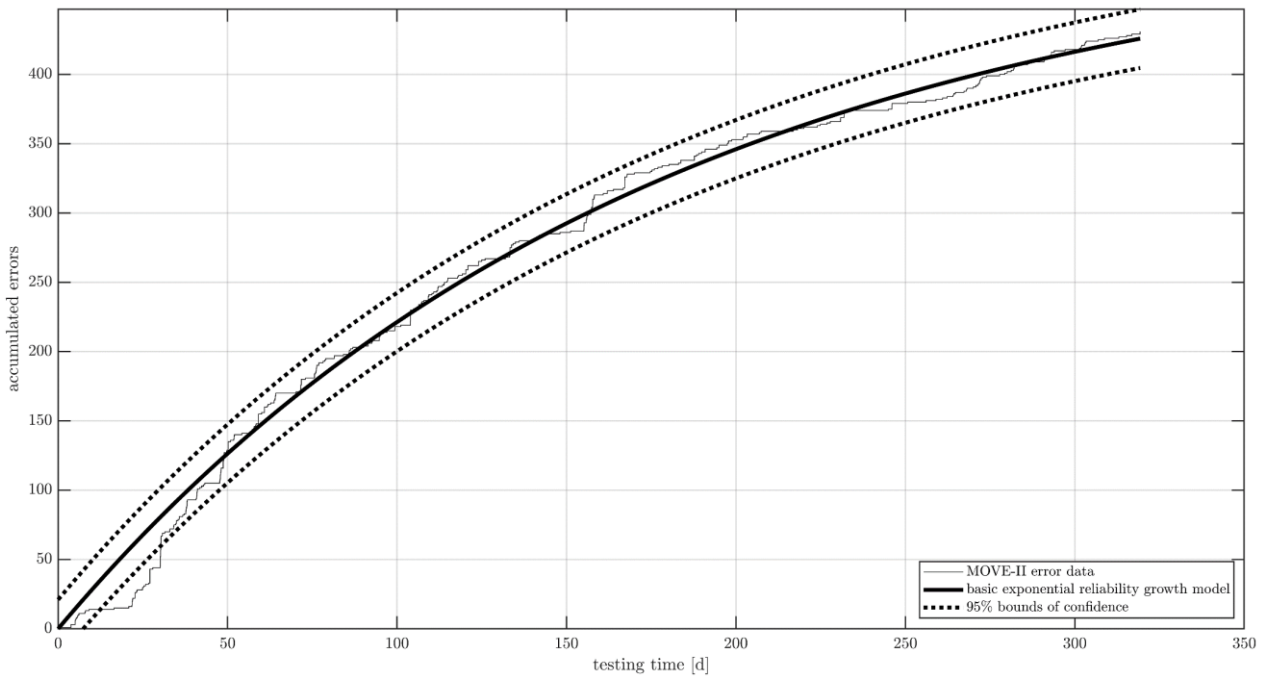


Figure 4-122: Fit of the basic exponential reliability growth model to the data of cumulative errors of the space segment of MOVE-II.

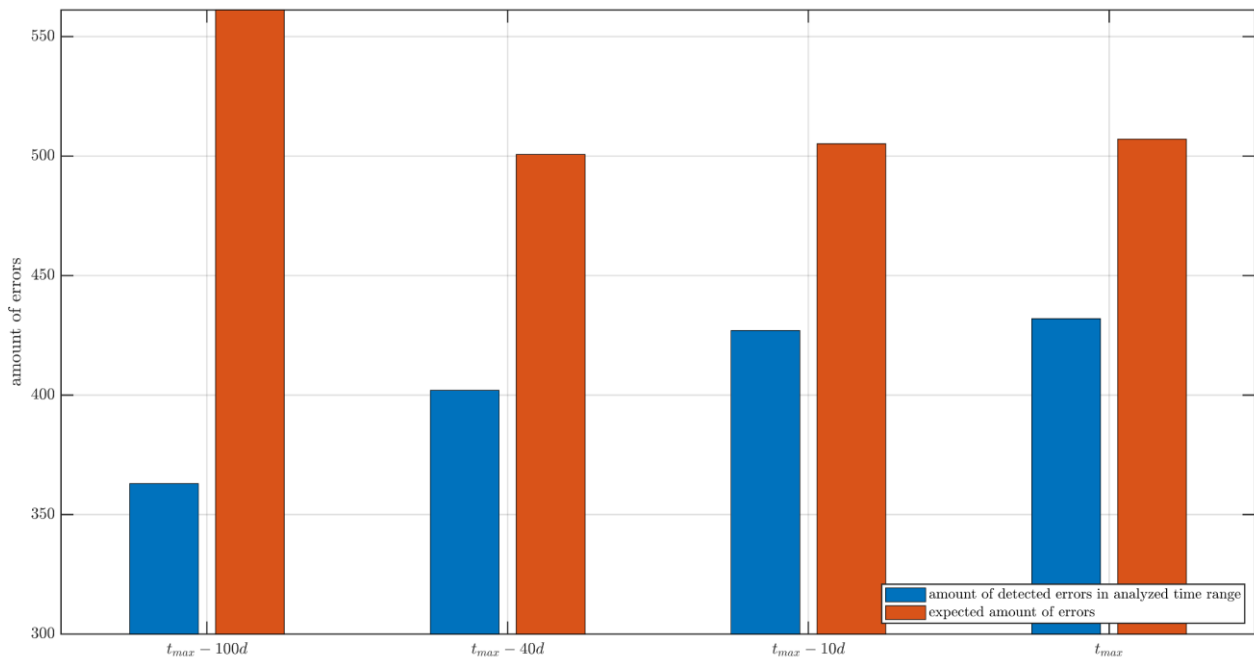


Figure 4-123: Stability of the prediction of the basic exponential reliability growth model when going back to earlier points in time.

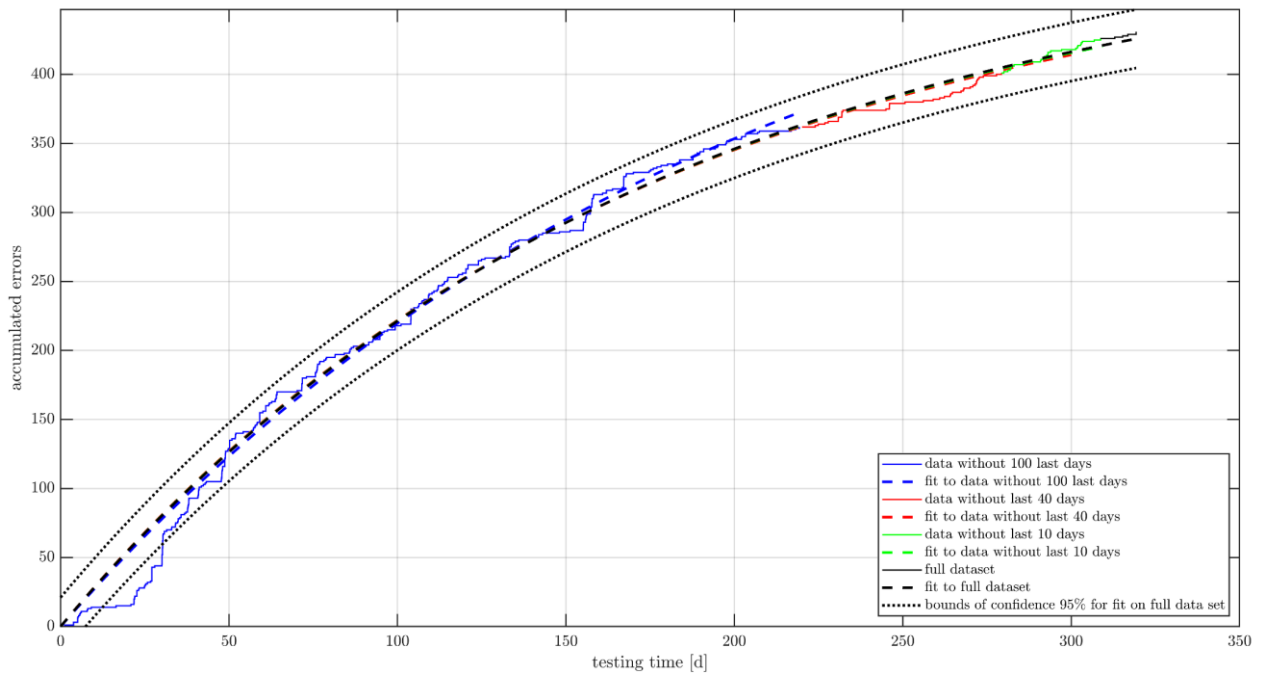


Figure 4-124: Prediction of basic exponential model with different time ranges. Blue depicts all data up to day 219, red all data up to day 279, green all data up to day 309 and black the full data set. The estimation of day 219 projected 54 more errors in the system than the other estimations.

Next, the model of Yamada & Osaki [180], based on a differentiation between easy- and difficult-to-detect errors, was applied to the data. They define the number of estimated errors up to a time t as:

$$H(t) = a \cdot \sum_{i=1}^2 p_i \cdot [1 - \exp(-b_i \cdot t)] \quad (49)$$

and:

$$b_1 > b_2; p_1 + p_2 = 1 \text{ and } 0 < p_i < 1 \quad (50)$$

Thus, it is a modification of the before used basic exponential model, and introduces 2 types of errors that can be categorized by the test personnel [180]. However, this categorization was not possible within MOVE-II and will be difficult to achieve in general by other university CubeSat teams in the view of the author. The reason for this is mainly the lack of experience of the involved people in such projects. For completeness, the model was applied to the MOVE-II data but showed enormous dispersion of its 95% confidence intervals on each estimated parameter. The model estimates a total number of 507.2 errors in the system, and thus is very close to the estimation of the basic exponential model. The basic fit can be seen in Figure 4-125. The sensitivity analysis showed a similar behavior as the basic exponential model, with more than 550 errors estimated at day 219, and around 500 errors estimated at the two points in time after that, as can be seen in Figure 4-126. Overall, the bigger dispersion of all parameters and the missing categorization induced us to neglect the results of this model.

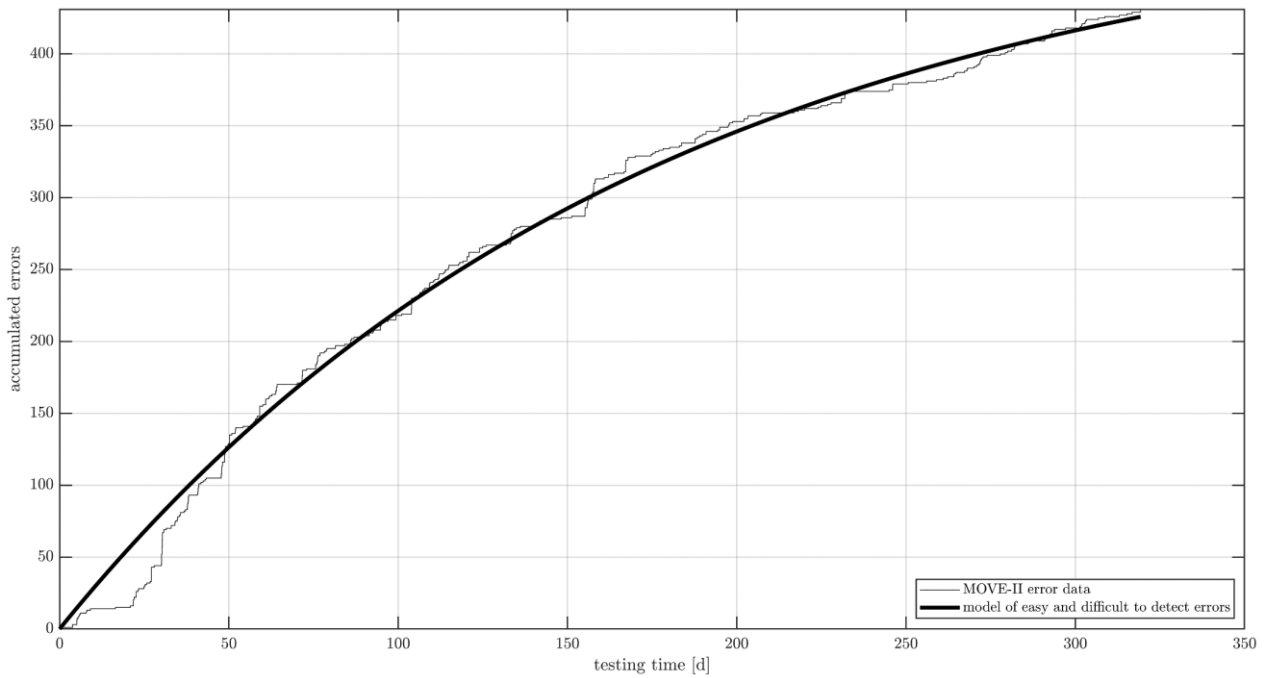


Figure 4-125: Fit of the Yamada & Osaki reliability growth model to the data of cumulative errors of the space segment of MOVE-II. 95% confidence intervals not shown due to large dispersion.

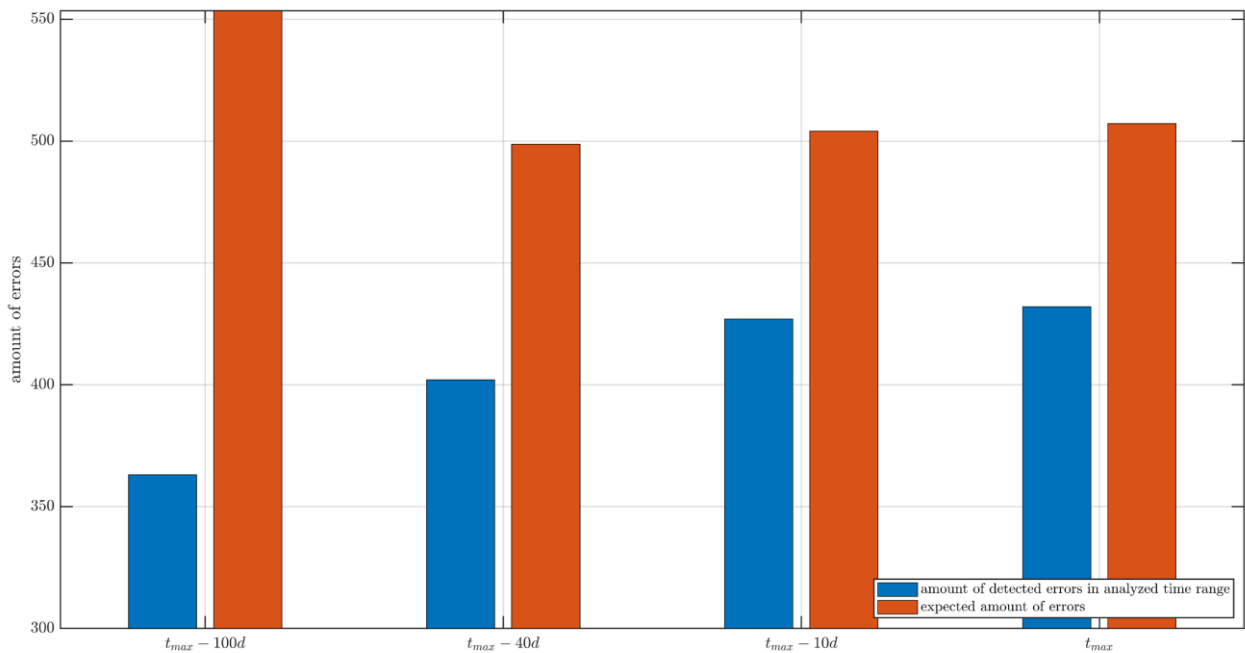


Figure 4-126: Stability of the prediction of the Yamada & Osaki reliability growth model when going back to earlier points in time.

The staggered entry of failure data for many subsystems as well as the space segment led to an investigation about reliability growth models in which a delayed starting time can be incorporated. The Master's Thesis of Florian Schummer [262], supervised by the author of this thesis, showed an exponential reliability growth model with variable starting date as a relatively easy adaption of the already presented basic exponential model. Based on the work of Ohba [179], the exponential model was modified to:

$$H(t) = a \cdot (1 - \exp[-b \cdot (t - t_0)]) \quad (51)$$

with:

$$a, b \text{ and } t_0 \geq 0 \quad (52)$$

The delayed starting date in the model can then be chosen specifically on the underlying error data or left as an additional parameter to be fitted. In our case, we left the starting date as a variable and thus expanded the exponential model to three parameters. As shown in Figure 4-127, the new model shifts the starting date to $t_0 = 7.8$ days (95% confidence interval: 6.8 days, 8.7 days). This results in an overall better fit of the staggered entry, and in a reduced number of total errors estimated in the system. The model estimates a total amount of 477.1 errors in the system, with a 95% confidence interval of 471.5 errors and 482.8 errors. As depicted in Figure 4-128, this estimation is within 20 errors, thus relatively constant, when going back in time. At $t = 219$ days, 478 errors are estimated by the model. Later in time this decreases to 460 errors at $t = 279$ days, but increases again at $t = 309$ days to 473 errors. The rate of errors detected per undetected error is estimated as 0.00684 per day, with a 95% confidence interval of 0.0066 per day and 0.0070 per day. Figure 4-129 depicts the prediction of the model with different time ranges.

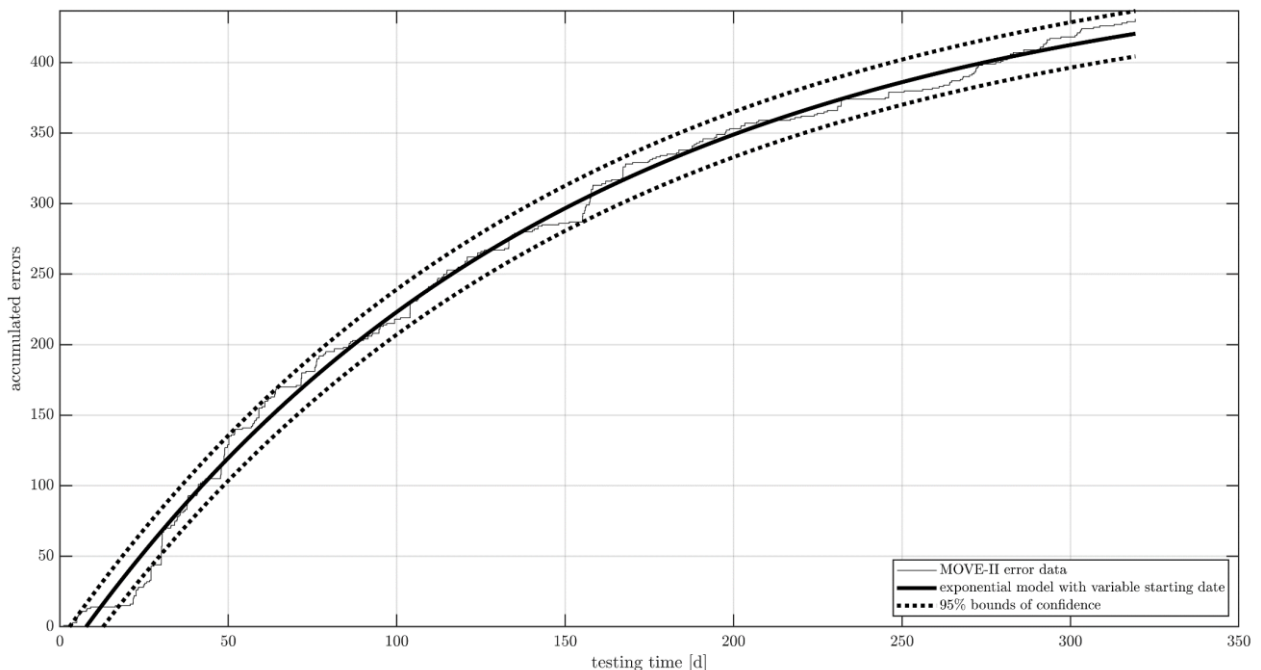


Figure 4-127: Fit of the exponential reliability growth model with variable starting date to the cumulative errors of the space segment of MOVE-II.

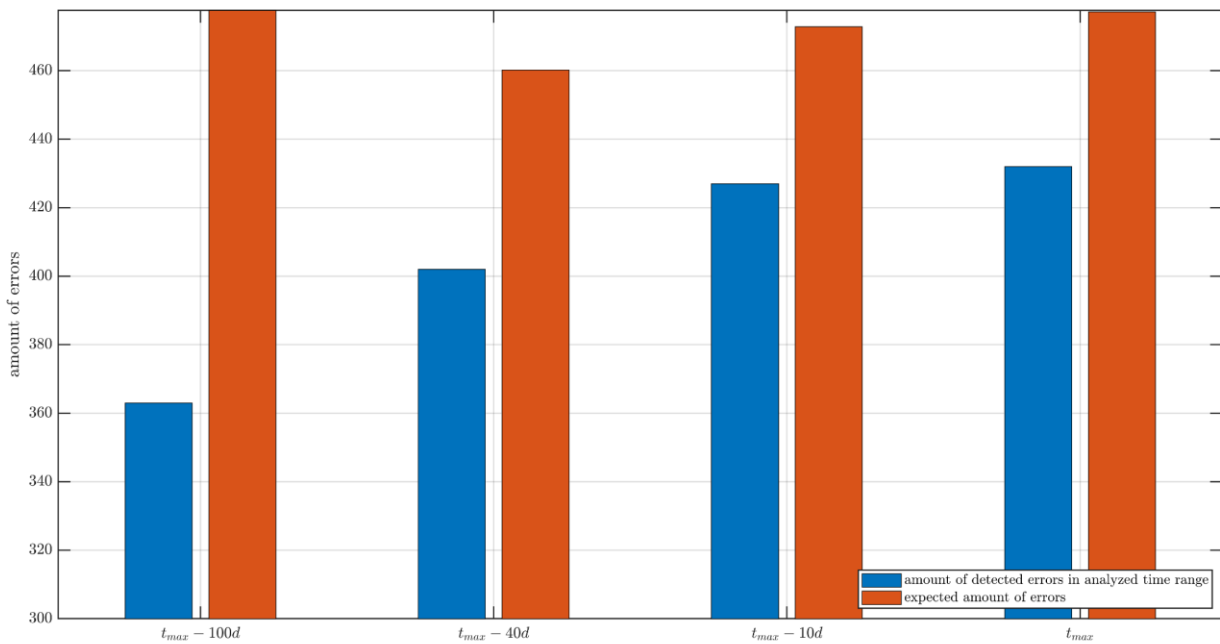


Figure 4-128: Stability of the prediction of the exponential reliability growth model with variable starting date when going back to earlier points in time.

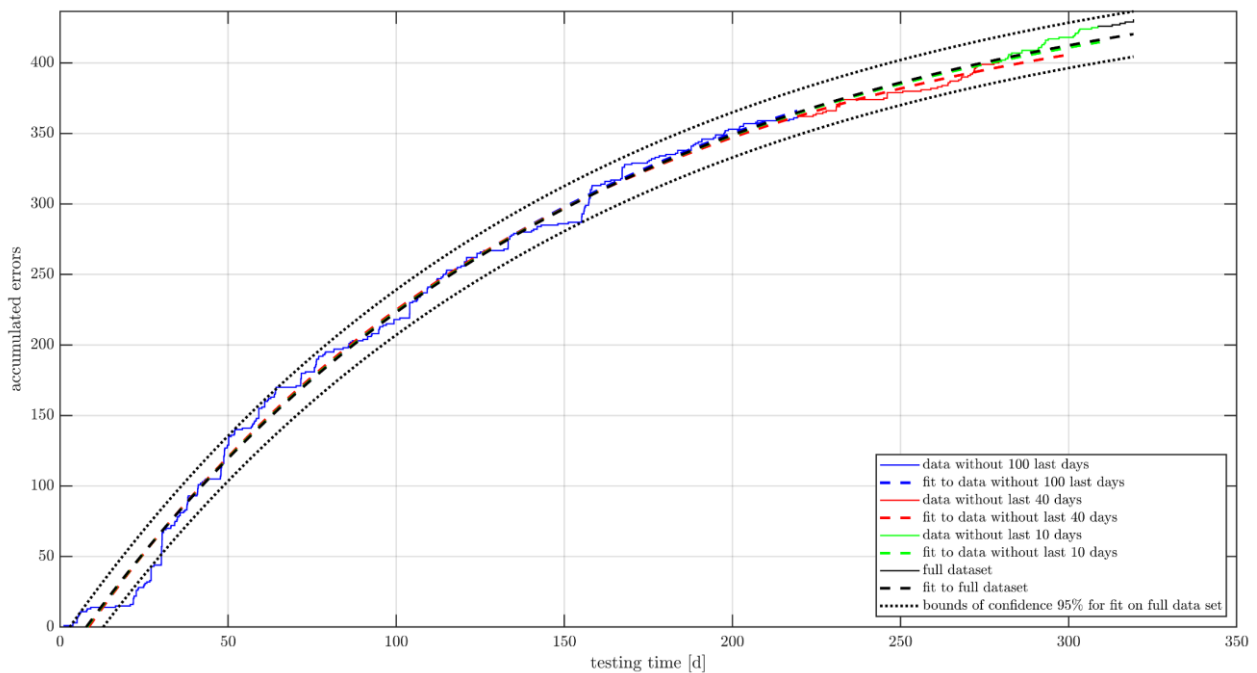


Figure 4-129: Prediction of the exponential reliability growth model with variable starting date using different time ranges. Blue depicts all data up to day 219, red all data up to day 279, green all data up to day 309 and black the full data set. The maximum difference of the expected amount of errors is no more than 18 errors for all estimations.

The last two models applied on the data of MOVE-II were so-called S-shaped models. S-shaped models are another way of dealing with a staggered entry of cumulative errors over time. As in many other projects under development, severe problems occurring right at the beginning of the system level testing can prevent the system from working at all, and thus stop the accumulation of errors for a certain amount of time. To

resolve that, Yamada, Ohba & Osaki [295] used a delayed S-shaped software reliability growth model that estimated the number of errors up to a time t as:

$$H(t) = a \cdot [1 - (1 + b \cdot t) \cdot \exp(-b \cdot t)] \quad (53)$$

With a not constant error detection rate per undetected error of:

$$d(t) = \frac{b^2 \cdot t}{1 + b \cdot t} \quad (54)$$

Thus, different variants of a staggered (or not staggered entry) can be modelled by the delayed S-shaped approach. Figure 4-130 (left) shows the number of estimated errors for different error detection rates and Figure 4-130 (right) the corresponding error detection rate.

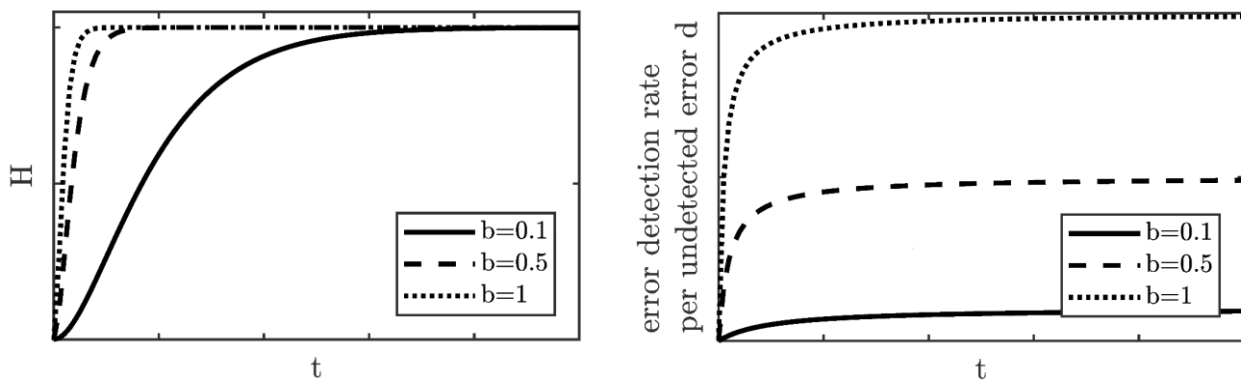


Figure 4-130: Number of estimated errors over time for different error detection rates of the delayed S-shaped software reliability growth model (left) and different error detection rate per undetected error $d(t)$ over time (right). Image Source: [262]

Initially, the delayed S-shaped model showed good results for our data, as can be seen in Figure 4-131.

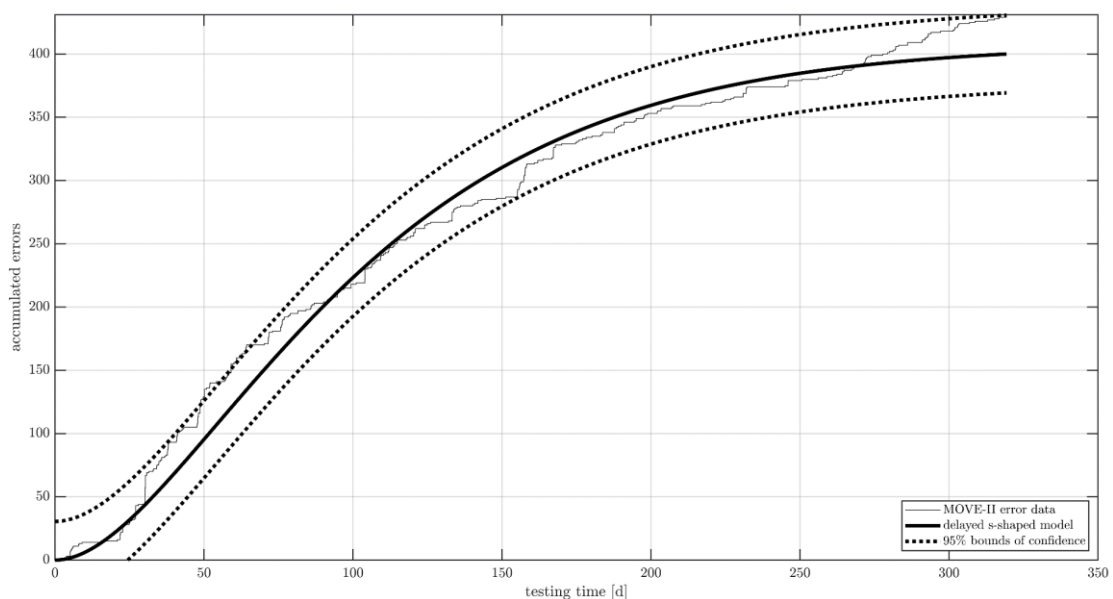


Figure 4-131: Fit of the delayed S-shaped software reliability growth model to the cumulative errors of the space segment of MOVE-II.

However, the total number of errors estimated by the model is 408 errors (95% confidence interval: 404 errors, 412 errors), which is below the actual number of found errors at day 319. This behavior of the model also showed when using earlier points in time, as depicted in Figure 4-132. With an exception at $t = 219$ days, the model under-estimates the amount of errors in the system at any point in time, which also can be seen in Figure 4-132.

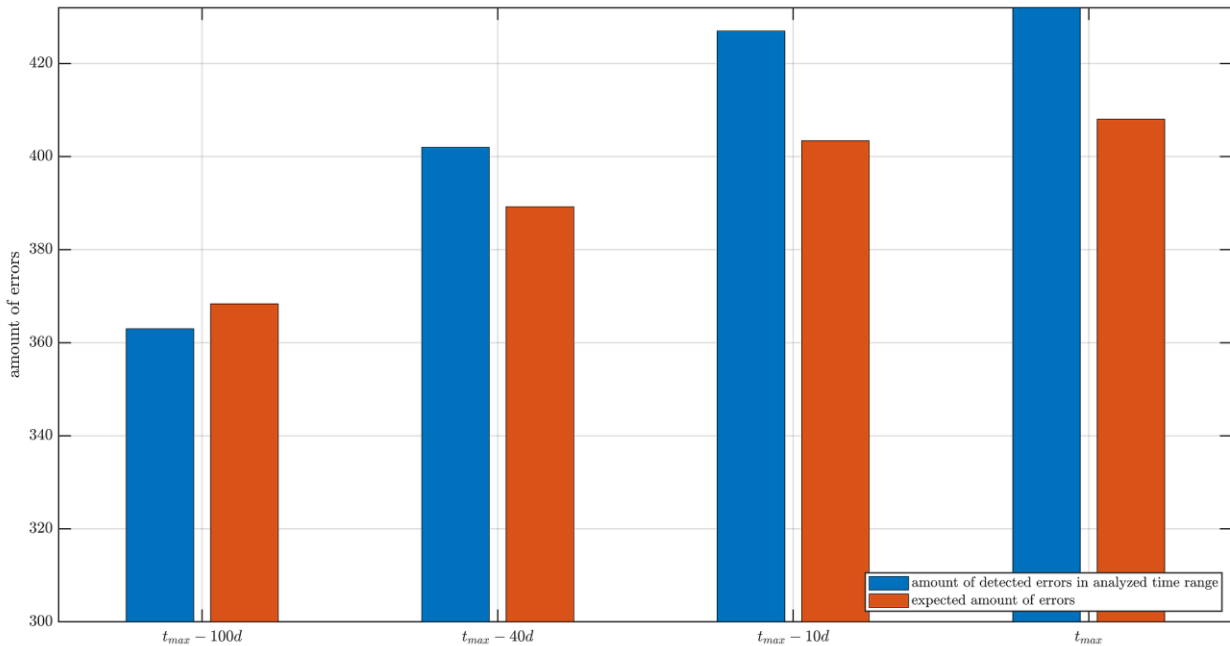


Figure 4-132: Stability of the prediction of the delayed S-shaped software reliability growth model when going back to earlier points in time.

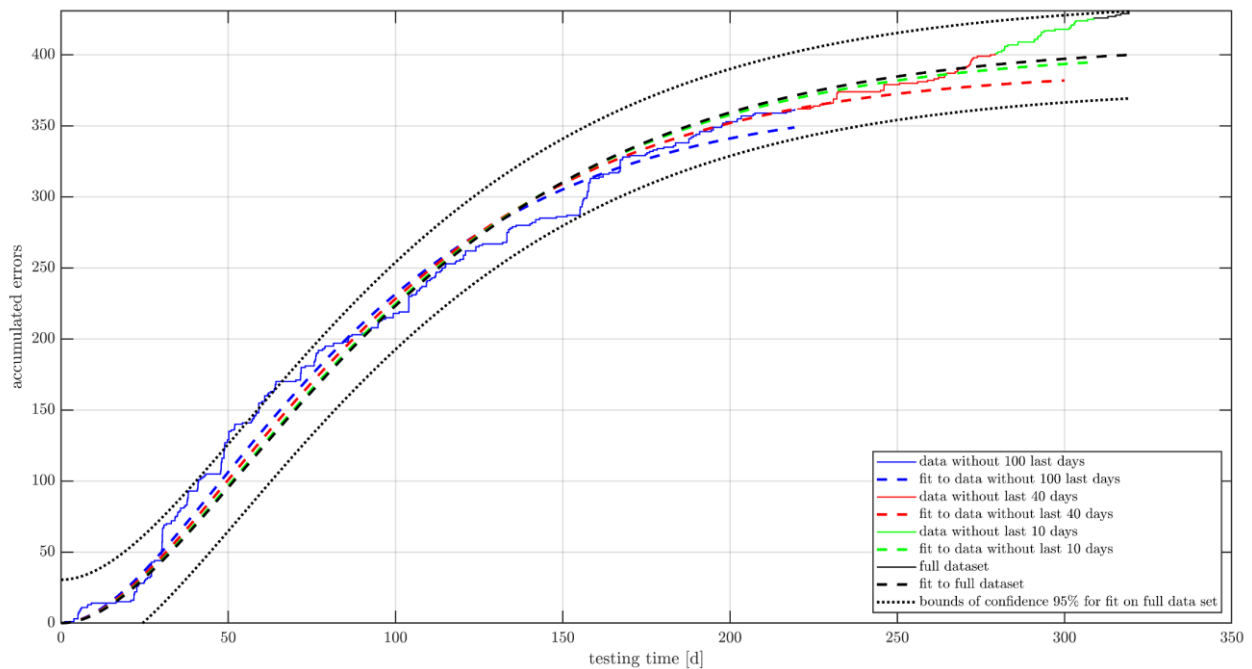


Figure 4-133: Prediction of the delayed S-shaped software reliability growth model using different time ranges. Blue depicts all data up to day 219, red all data up to day 279, green all data up to day 309 and black the full data set. The estimation of total errors in the system is increasing as the test proceeds and mostly underestimates the number of errors in the system.

This underestimation could originate from several sources, since the delayed S-shaped model is based on several assumptions, as presented by Ohba [179]. An example for an assumption of the model that is not entirely fulfilled in MOVE-II is, that detected faults can be entirely removed. Some errors that occurred in MOVE-II that are not critical had to remain in the system, as they were too complicated to be fixed or only caused very limited performance decline. Also Ohba states that the accuracy of the model decreases if the time delay between detection and correction is not negligible and the test effort for fault detection and correction is not constant [179]. Both characteristics are partly fulfilled in MOVE-II, and we will further discuss this in Chapter 5.

As the last model, the inflection S-shaped model, based on the work of Ohba [179], was used. The model is based on the logistic curve model, and describes a software failure detection phenomenon in which the more failures are detected in the error detection process, the more undetected failures become detectable [179]. It estimates the number of errors up to a time t as [180]:

$$H(t) = \frac{a \cdot [1 - \exp(-b \cdot t)]}{[1 + c \cdot \exp(-b \cdot t)]} \quad (55)$$

with a not constant error detection rate per undetected error of [180]:

$$d(t) = \frac{b}{[1 + c \cdot \exp(-b \cdot t)]} \quad (56)$$

The so-called inflection parameter c is defined as [179]:

$$c = \frac{1 - r}{r}, \quad r > 0 \quad (57)$$

where r is the inflection rate of the function, which describes the ratio of number of detectable errors to the total number of errors. If r becomes one, all errors are detectable from the beginning of the test and the model is equivalent to the exponential model. Towards zero, the model approaches the logistic curve, which represents a system in which only a few errors are detectable at the beginning and become rapidly detectable afterwards [179].

Similar to the delayed S-shaped model, the inflection S-shaped model showed an overall good alignment to the MOVE-II space segment error data, as depicted in Figure 4-134. A total number of 453.9 errors (95% confidence interval: 443.8 errors and 464.1 errors) is estimated by the model and the goodness-of-fit is $R^2 = 0.9935$. The function is in between an exponential and a logistic curve, as the inflection rate is $r = 0.596$ (95% confidence intervals: 0.526, 0.665). However, as the delayed S-shaped model before, the model shows a constantly increasing prediction of total errors, making it unreliable for predictions of remaining errors left in the system over time (see Figure 4-135). This is also shown in Figure 4-136, in which the estimations increase with time. Different from the delayed S-shaped model, the model is never below the current number of detected errors with its estimation, as also can be seen in Figure 4-136. Nevertheless, this increasing estimation of errors over time makes both S-shaped models inferior to the presented exponential models, in which the estimated errors decrease over time or stay relatively constant. It seems better to overestimate the total number of errors left in the system and correct that number later in the process as to underestimate it and stop testing too early.

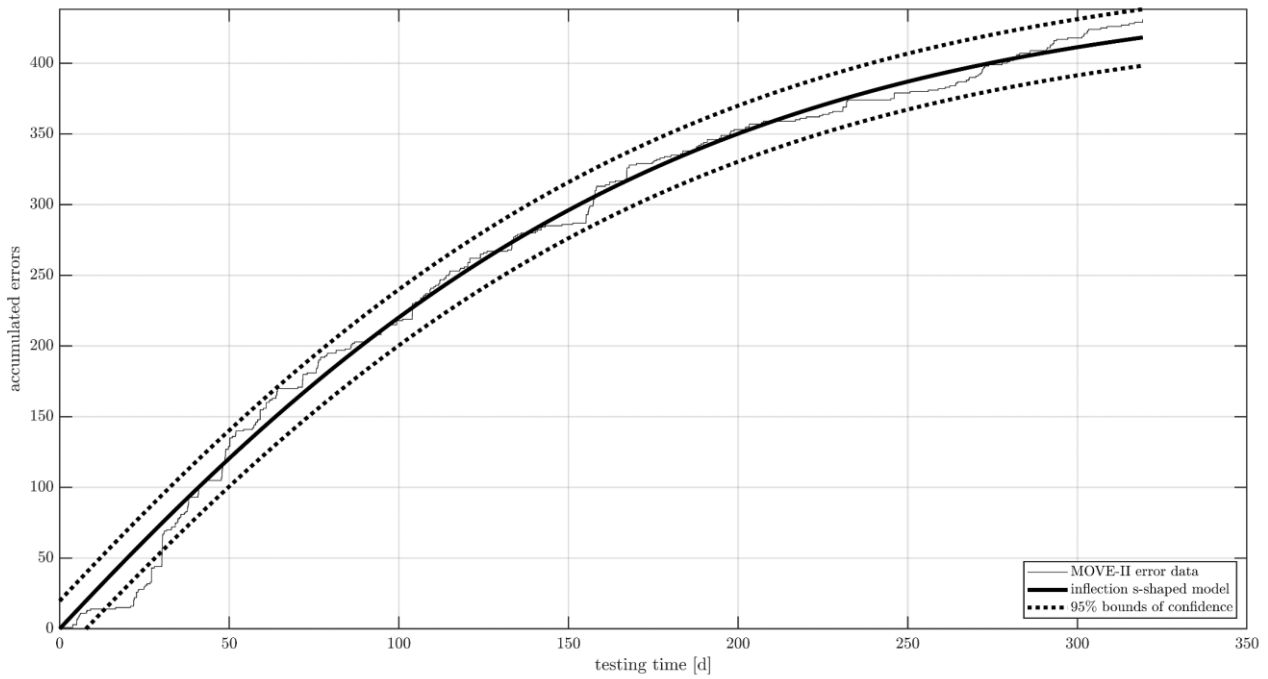


Figure 4-134: Fit of the inflection S-shaped software reliability growth model to the cumulative errors of the space segment of MOVE-II.

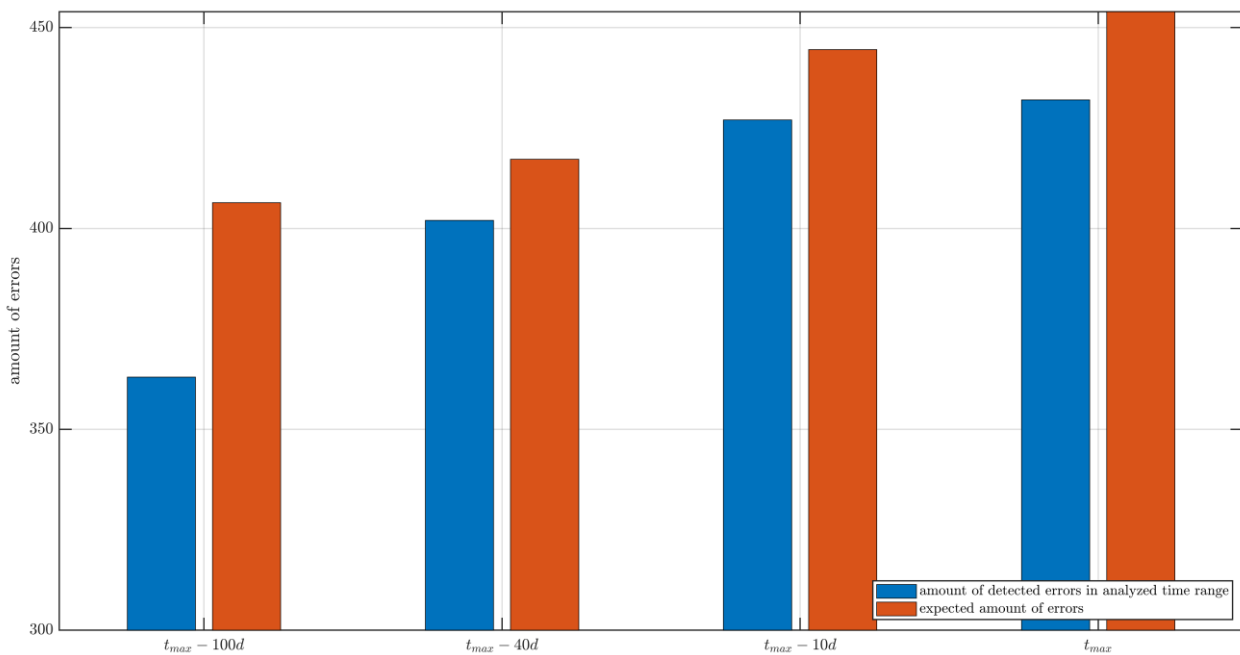


Figure 4-135: Stability of the prediction of the inflection S-shaped software reliability growth model when going back to earlier points in time.

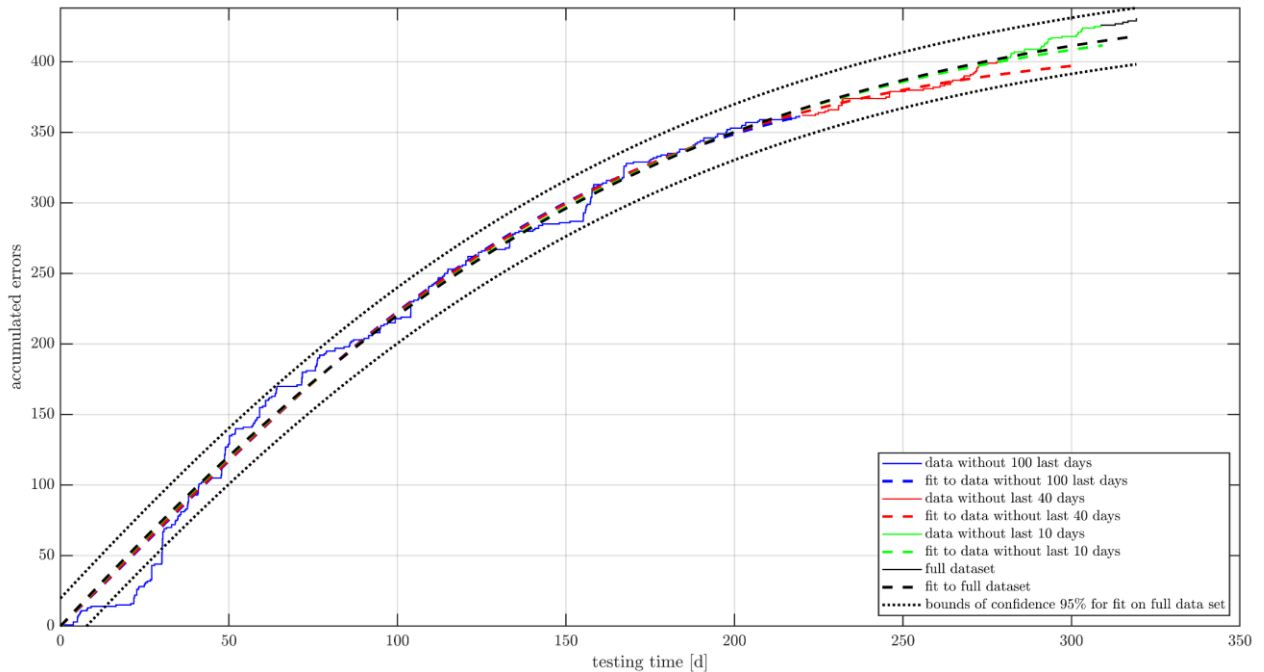


Figure 4-136: Prediction of the inflection S-shaped software reliability growth model using different time ranges. Blue depicts all data up to day 219, red all data up to day 279, green all data up to day 309 and black the full data set. The estimation of total errors in the system is increasing as the test proceeds.

To summarize, the basic exponential model and the exponential model with variable starting date showed the best results for our data, as can be seen in Figure 4-137. As already pointed out, the model of Yamada & Osaki of easy- and difficult-to-detect errors was neglected, since it showed a big dispersion of all parameters, which might have originated from the missing categorization of the errors in MOVE-II. Both S-shaped models showed an increasing estimation of the total amount of errors over time, which is inferior to the other models. The results of the analysis of the overall failure rate of the space segment of MOVE-II is further discussed in Chapter 5.

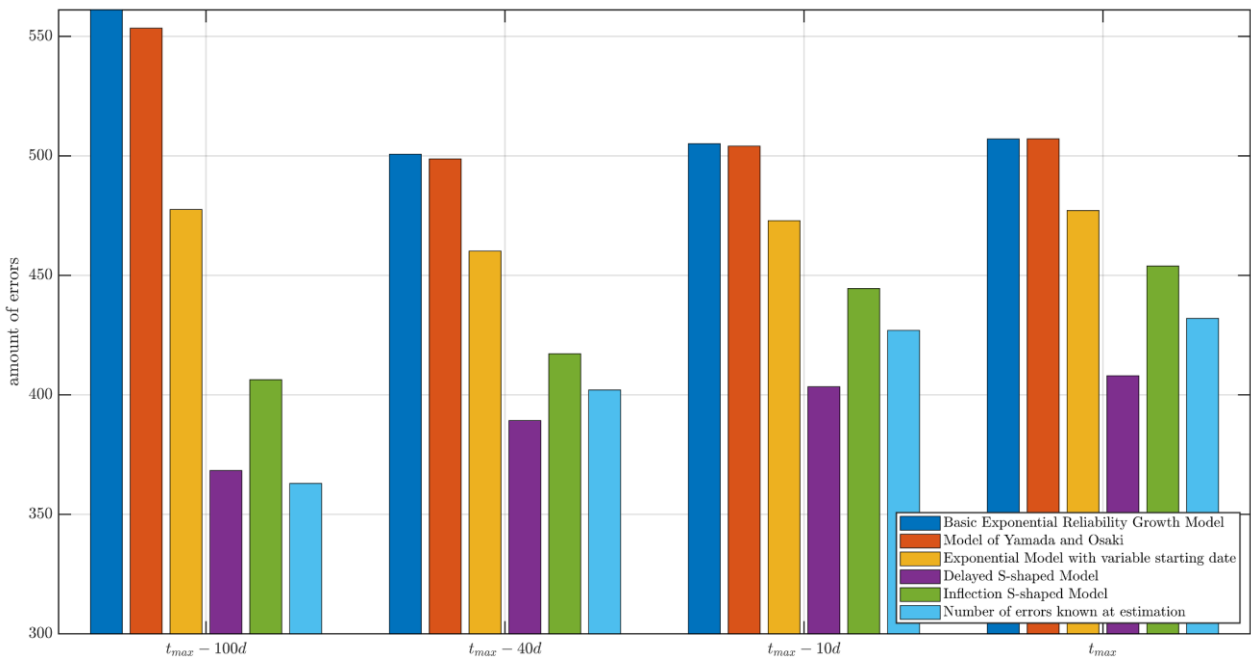


Figure 4-137: Comparison of the estimation of all models and the number of known errors at different points in time for all errors on the space segment of MOVE-II.

As described earlier, the MOVE-II FRACAS was also filtered regarding critical failures that had a great chance of stopping the mission in space. 113 of these failures were found, and reliability growth modelling was also applied to this sub-group of errors. Overall, the basic exponential model and the exponential model with variable starting date again showed the best characteristics of all growth models for this sub-group of failures. The results of both models and a sensitivity analysis with varying length of observation are presented in the following. The missing distinction between easy- and hard-to-fix failures in the database resulted again in not trustworthy results of the Yamada & Osaki model. As in the analysis with the complete dataset from the space segment, the S-shaped models predicted an increasing number of failures with increasing time, while underestimating the total number of critical failures for each prediction. Thus, the results of the Yamada & Osaki model as well as both S-Shaped models are summarized in Figure 4-144, and depicted in Appendix B, Figure 6-7 to Figure 6-15.

The basic exponential model foresees a total amount of 116.8 critical errors in the MOVE-II space segment (95% confidence bounds: 115.3 critical errors, 118.4 critical errors), which is 4 errors more (rounded value) than found in the system until the observation window ended. As seen with the complete dataset from the space segment, the model overestimates the total amount of critical errors at earlier points in time. Figure 4-139 depicts this overestimation, which starts at 119.2 critical errors ($t = 219$ days) and then decreases slightly to 117.2 critical errors ($t = 279$ days) to 117 critical errors ($t = 309$ days). This overestimation is not as critical as the aforementioned underestimation by the S-shaped models, as the system level testing would then be planned for a longer period of time than necessary. However, this imposes not only an unnecessary growth of resources and time spent in system level testing, it also could pose the risk of missing a launch opportunity if the spacecraft is declared not ready for launch. In an extreme case, a project could also see termination if too many critical errors are estimated. The results of the predicted critical errors found for $t = 319$ days is remarkably close (112.9 critical errors) to the actual number of critical errors found until that point in time (with a 95% confidence interval of 102.7 critical errors and 123.2 critical errors). The goodness-of-fit of the function is $R^2 = 0.9744$.

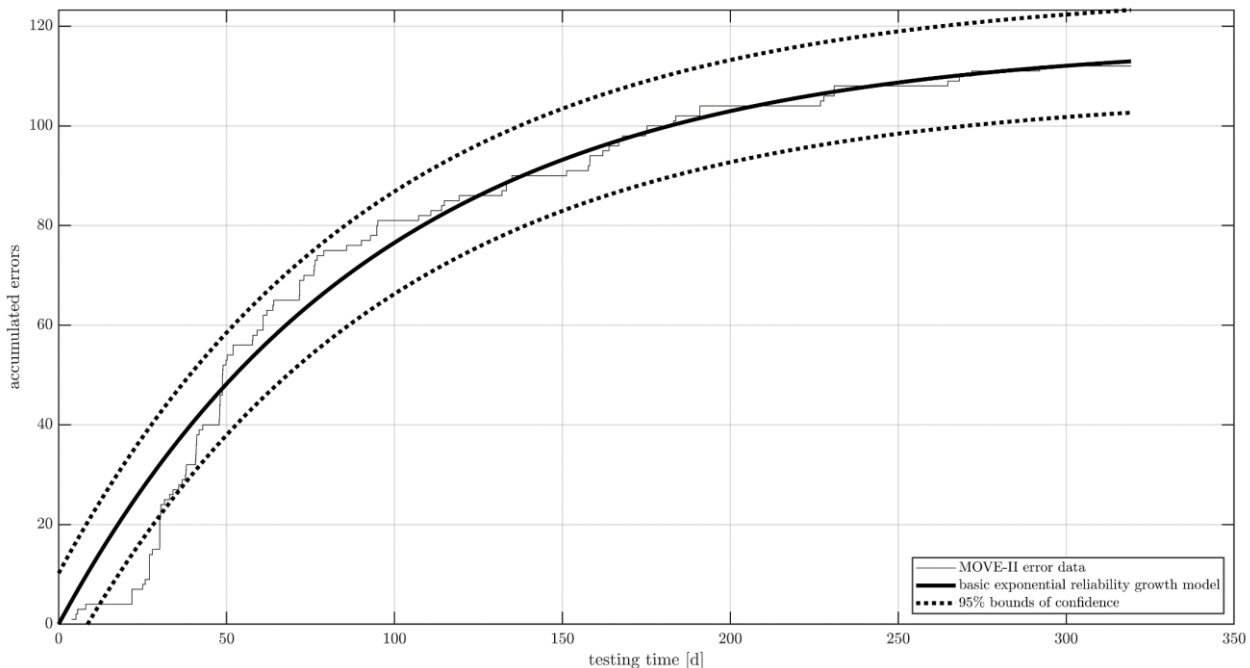


Figure 4-138: Fit of the basic exponential reliability growth model to the critical failures of the space segment of MOVE-II.

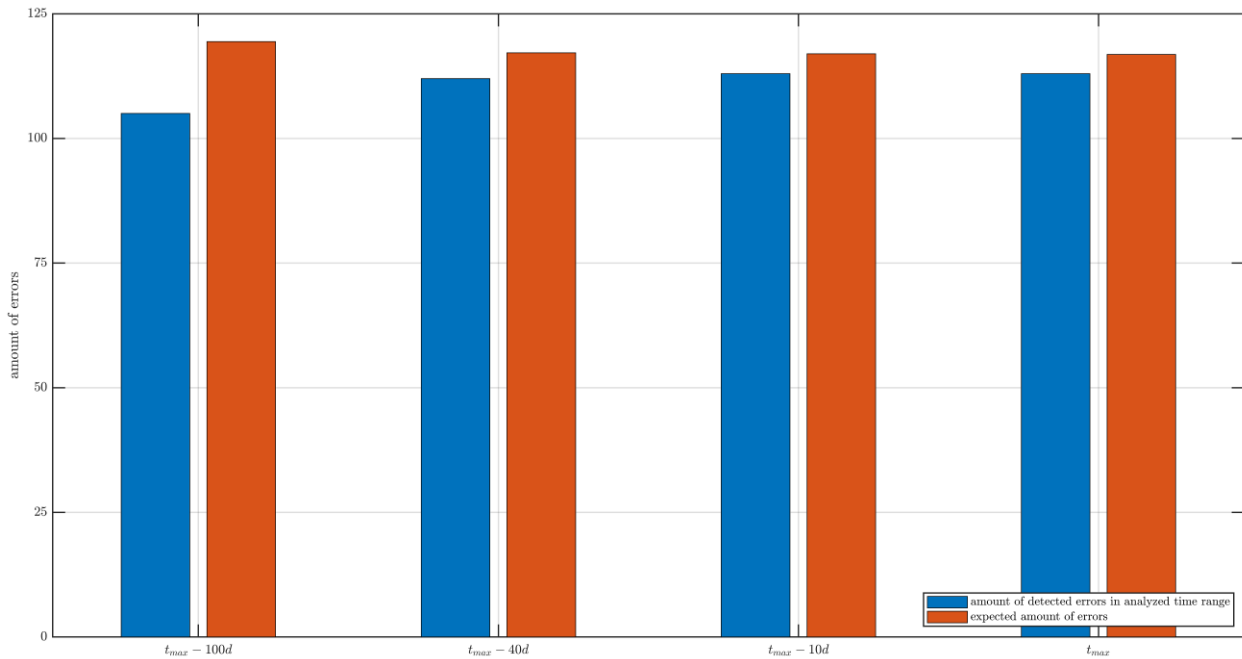


Figure 4-139: Stability of the prediction of critical failures of the space segment of MOVE-II by the basic exponential reliability growth model when going back to earlier points in time.

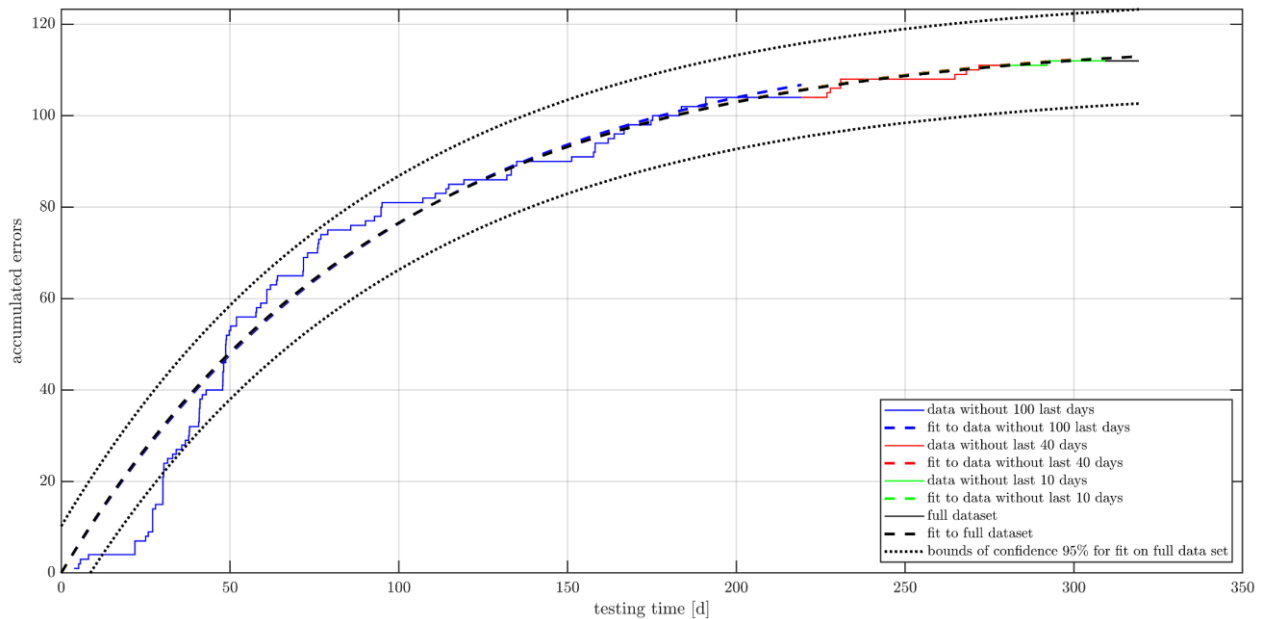


Figure 4-140: Prediction of critical failures of the space segment of MOVE-II by the basic exponential reliability growth model using different time ranges. Blue depicts all data up to day 219, red all data up to day 279, green all data up to day 309, and black the full data set.

The second model studied was the exponential model with variable starting date and as before it showed the overall best prediction results, as can be seen in Figure 4-141 and Figure 4-142. A total number of 112.9 critical errors are estimated by the model, which is exactly the number of critical errors found in the system at the end of the observation window (rounded value). The 95% confidence intervals of this estimation are 111.9 critical errors and 113.9 critical errors. The delayed entry is estimated with $t_0 = 9.5$ days (95% confidence interval: 8.5 days, 10.5 days), which is close to the value estimated by the model when using a full dataset. The sensitivity analysis showed that the model is stable when using past data, estimating 110 critical errors ($t = 219$ days), 111.8 critical errors ($t = 279$ days) and 112.7 critical errors ($t = 309$ days). Thus, the estimation of the total failures was within a range of only 3 critical errors when using data of the last 100

days of the observation window, as depicted in Figure 4-143 and Figure 4-144. The model estimated a number of 111 critical errors found at $t = 319$ days, with a 95% confidence interval of 118.8 critical errors and 103.2 critical errors. As already pointed out, the number of critical errors found at that point in time was the overall number predicted by the model (113). Despite this, system level testing in MOVE-II was continued in order to increase confidence in the system as well as the prediction models and thus minimize the 95% confidence interval of the prediction. If critical errors are found in the future, the growth models will be updated, and the project plan adapted accordingly.

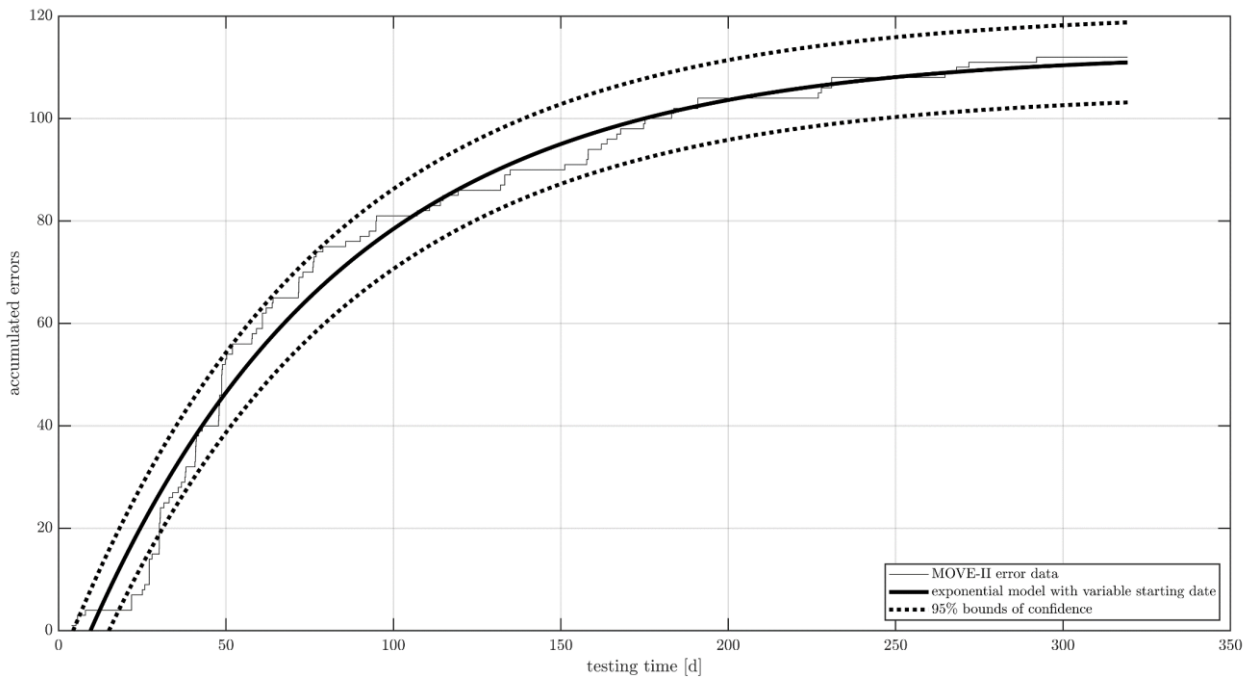


Figure 4-141: Fit of the exponential model with variable starting date to the critical failures of the space segment of MOVE-II.

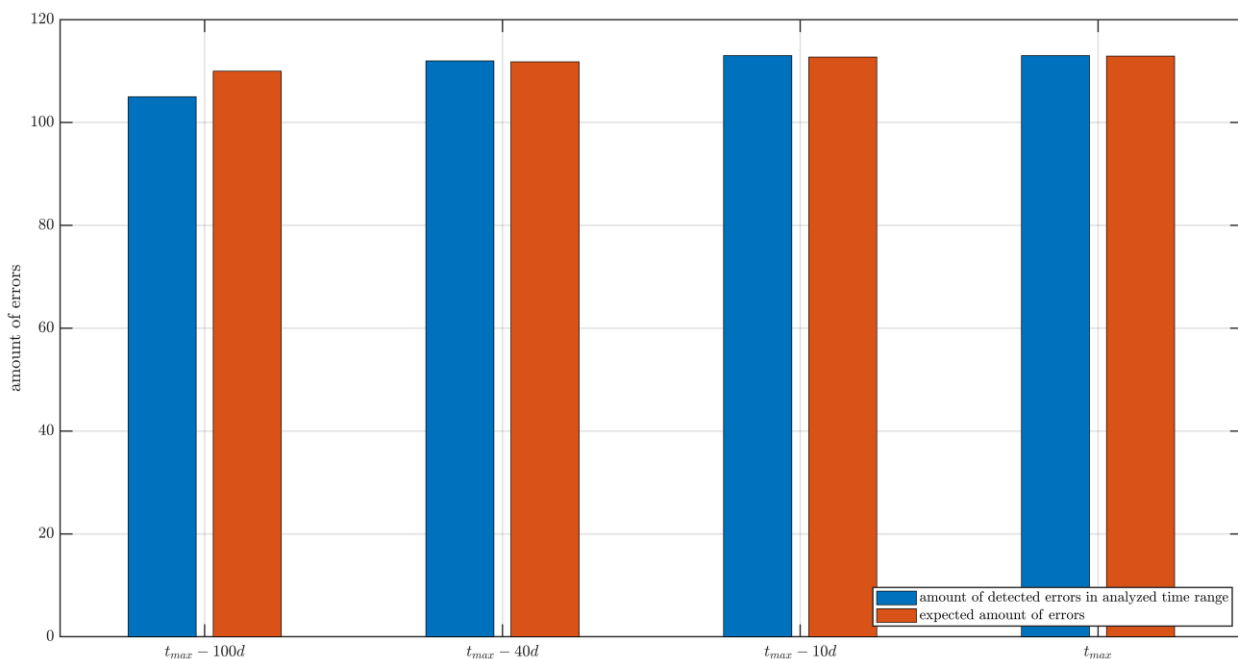


Figure 4-142: Stability of the prediction of critical failures of the space segment of MOVE-II by the exponential model with variable starting date when going back to earlier points in time.

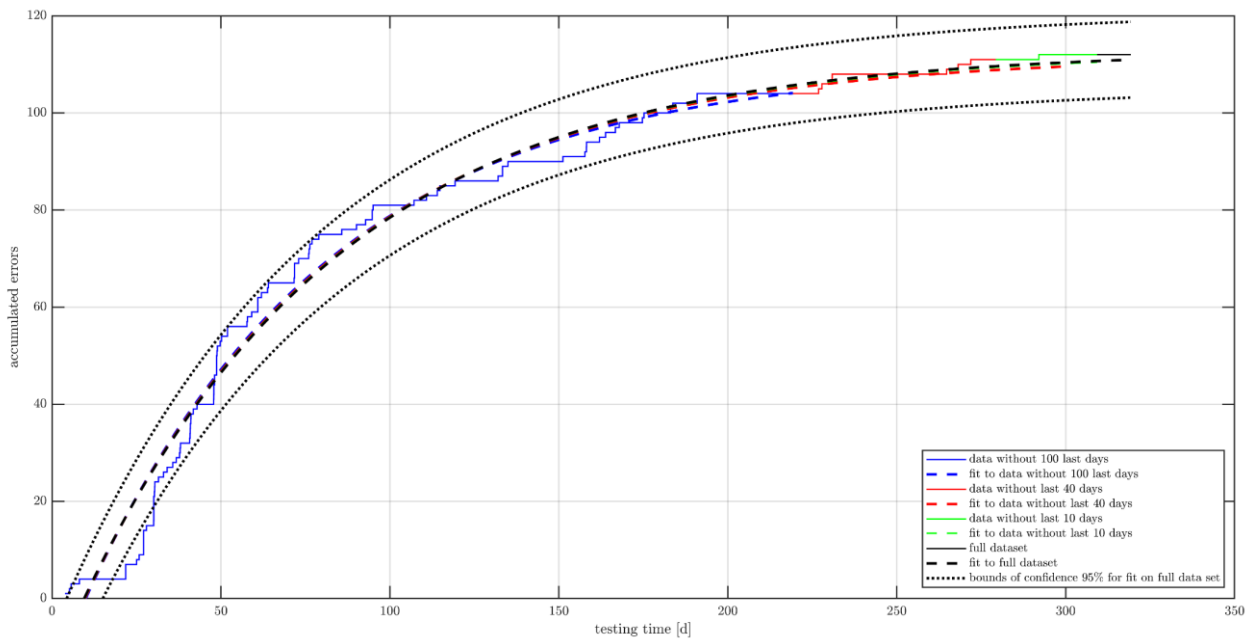


Figure 4-143: Prediction of critical failures of the space segment of MOVE-II by the exponential model with variable starting date using different time ranges. Blue depicts all data up to day 219, red all data up to day 279, green all data up to day 309, and black the full data set.

Overall, the exponential model with variable starting date seems to fit best for our purposes. It shows, that the number of critical errors remaining in the system at the end of the observation window is below 0.5, thus limiting the risk of an occurrence of such a failure in space. However, as noted before, the onset and correction of failures is always linked to the tests done. Thus, if certain tests are not done and the errors are never provoked, they might occur in space nevertheless statistical data do not predict it. At the end of the day, confidence in the system must be achieved by testing and not by statistical models. Figure 4-144 summarizes the results of the different models for the analysis of critical failures in the MOVE-II space segment.

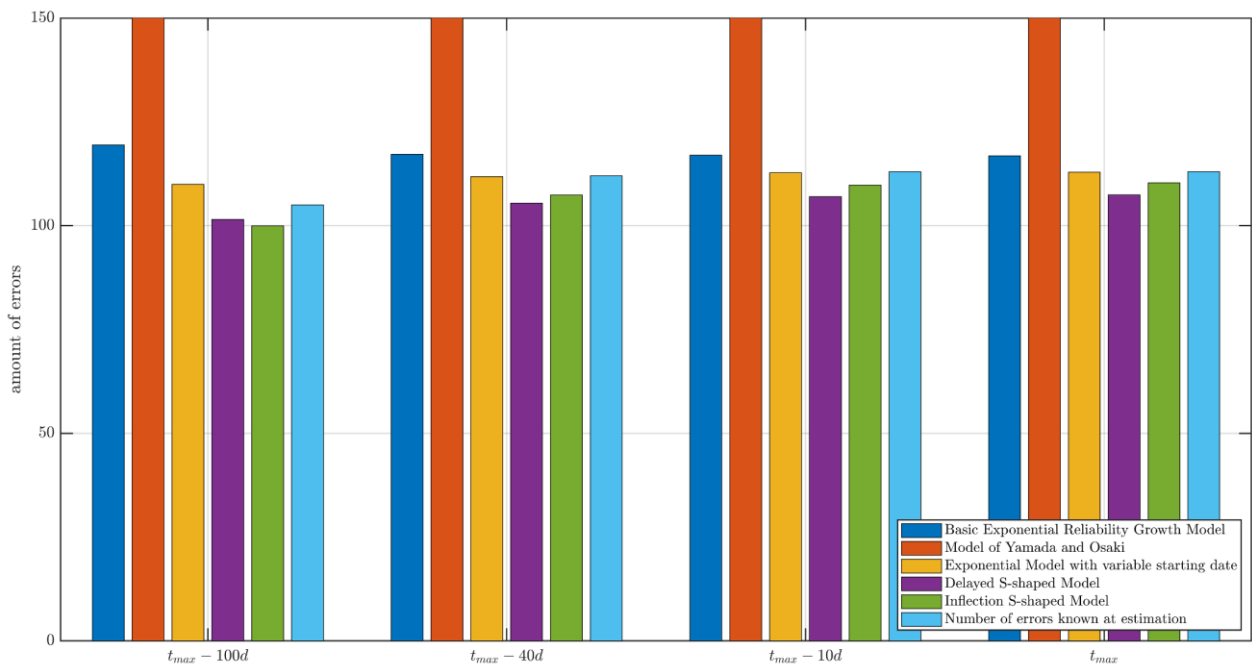


Figure 4-144: Comparison of the estimation of all models and the number of known errors at different points in time for all critical errors on the space segment of MOVE-II.

To summarize the subsection on the reliability assessment of MOVE-II, we have seen that the combination of an easy-to-use FRACAS system with a satellite test environment that allows remote access to the satellite are two keys to achieve exhaustive functional tests. The mindset of testing should be to uncover as many errors as possible during a test, not to prove that certain subsystems or software code is “passing” the test. Also, environmental tests should be incorporated in the assessment statistics since they often help uncovering failures of different root cause than system level functional tests. As already pointed out, the satellite (and its parts) must be qualified for space and launch environment. Thus, tests simulating those environments are inevitable. Nevertheless, as we have seen in Subsections 2.1.3 and 2.3.2, many CubeSats fail due to engineering flaws that could have been detected before launch, mostly when testing in a TLYF manner.

For CubeSats, limited functional testing has been proven as one of the major reasons for the high DOA and infant mortality rate. The shown reliability assessment models, foremost the exponential model with variable starting date and the basic exponential model, could help to achieve better management decisions regarding the duration and the extent of system level testing in CubeSat missions. MOVE-II, as a university-based project, shares many of its characteristics with other CubeSat projects worldwide, and the time needed for the overall project is not exceeding projects in similar environments. We maximized our time in system level testing, spending more than one year in that phase at the time of this writing. By doing this, we hope to minimize the chance of a critical failure occurring early into our mission to a reasonable value. In our case, the difference between critical failures found in the system and the prediction by the exponential growth model with variable starting date is only a fraction of one failure. For us, this means that we are ready for launch with our system, despite several not-critical failures still being in the space segment and predicted by the exponential models. Also, we will continue to work on our OPS and GS, since those two systems show no clear evidence of saturation so far. Overall, reducing the number of estimated critical errors to a value below a fraction of one minimizes the risk of a critical error still being hidden somewhere in the system. Nevertheless, not tested environments and testing vectors that were not considered pose always a remaining risk, and in our opinion parts of this risk cannot be eliminated. By maximizing the number of different persons interacting with the satellite, we tried to get as heterogeneous testing vectors as possible, but this is effort is decreased by the limited experience common in universities.

Finally, to transition slowly to our last subsection on results, dealing with reliability prediction, we can use data of our basic exponential reliability growth model to estimate the reliability of our system after testing ended. This is based on the basic concept, that any critical failure (or mathematically speaking, also a fraction of it) still in the system will cause a failure of the satellite in the future. Thus, reliability is calculated as the probability of no critical error occurring within the future observation window, based on the results of the basic exponential growth model. This of course is a great simplification of a more complex problem, as spacecraft reliability is not only dependent on results of reliability assessments, but also on the not known or not considered test vectors and the unique environment in space. Nevertheless, this estimation could be useful for program managers, not for its absolute values, but to see deviations over time and assess up until what point in time system level testing is till useful. For that purpose, the reliability can be estimated as [262]:

$$R(\Delta t, t) = \exp[a \cdot \{\exp(-b \cdot (\Delta t + t)) - \exp(-b \cdot t)\}] \quad (58)$$

Figure 4-146 shows the estimated reliability over time after system level test hypothetically ended. As aforementioned, this estimation should rather be seen from a functional testing point of view than from a complete “on-orbit” estimation. Despite its incompleteness, it gives a first estimation of the maturity of the system under test. Currently reliability prediction methods, as we have seen in Subsection 2.2.1, are assuming that the system is failure-free at start-of-life and that therefore the reliability can be calculated as the sum of the constant reliability rates of all parts (also taking redundancies into account if implemented).

With our approach, we focus on the incompleteness of most systems, as the reliability is predicted from the failures seen in system level testing. Thus, our approach is rather working with the early phase of the bathtub curve, concentrating on infant mortality, and not the constant failure rate that comes after that. For CubeSats this seems currently a valid approach since many systems are still stuck in the infant mortality region when launched and could be improved through more system level testing. As soon as the CubeSat world, especially those built in universities, matured and evidence becomes available that systems achieved to go past the infant mortality region, reliability prediction methods working with sums of all parts can be used. Today this is already the case for more experienced CubeSat teams, and it might also have a higher occurrence in commercial CubeSat missions than in university ones.

For that purpose, but also for the time in the future when most CubeSats arrived in the constant failure rate region, we also investigated reliability prediction methods suitable for resource-restricted projects such as ours, and the results of that will be presented in the next subsection. Finally, Figure 4-146 shows the reliability achievable at certain points in time after launch when continuing the system-level testing in MOVE-II, assuming that the same resources (time, persons) are spent. Such statistics could be useful for project managers to decide whether to continue their efforts or not. Of course, since CubeSats are always secondary payloads and often heavily restricted in resources, this is just one of many factors that will influence that decision, but in our opinion it is one that is worth to look at.

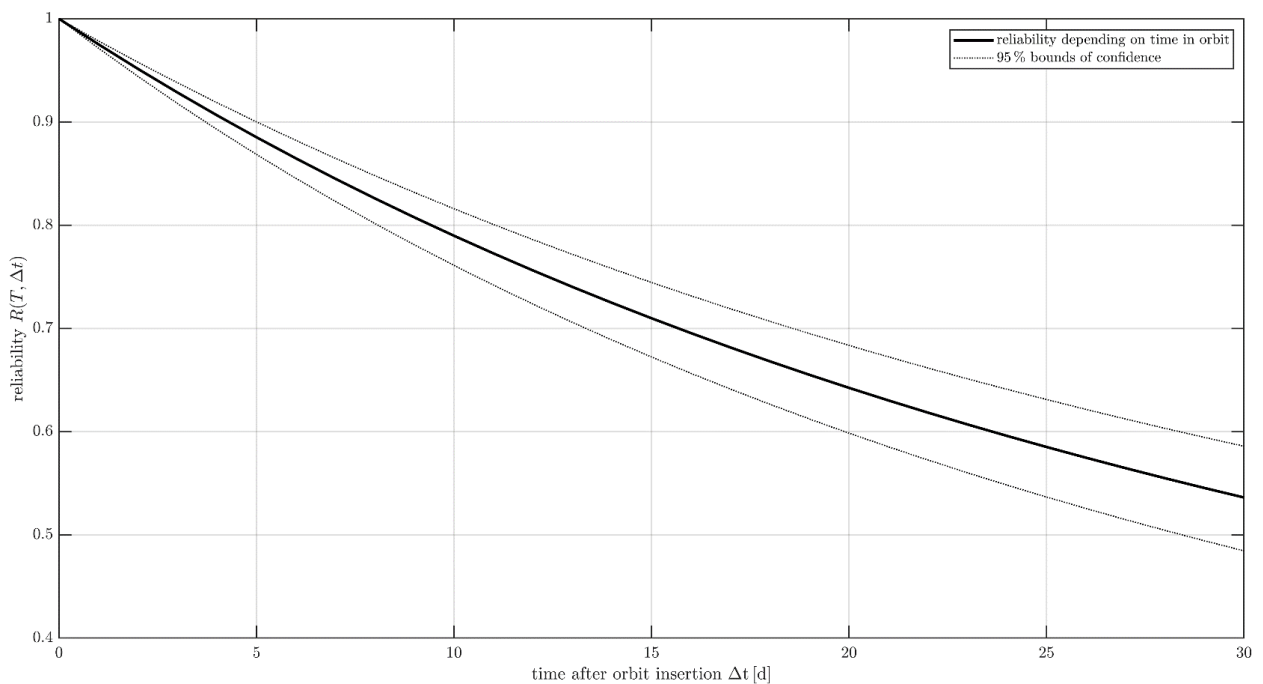


Figure 4-145: Reliability estimation of MOVE-II based on the number of critical failures estimated by the basic exponential reliability growth model.

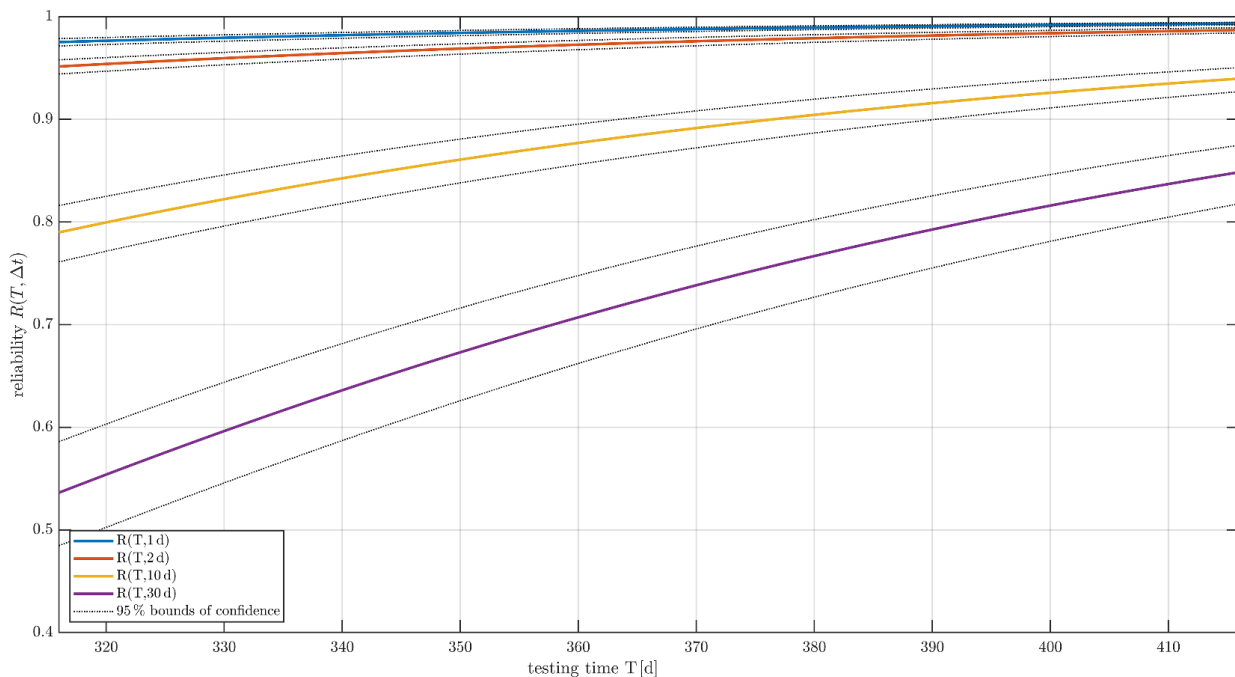


Figure 4-146: Estimated reliability if the system level tests would continue after the last day of the observation window (day 319). Blue depicts the reliability after day one, red after day two, yellow after day 10 and purple after day 30.

4.3.3 Predicting the Reliability of MOVE-II

This subsection is partially based on one conference paper [264] by the author of this thesis and the Master's Thesis of Michael Weisgerber [296], which was supervised by the author of this thesis.

As we have seen in Subsection 2.2.1, most reliability prediction methods work by summing up the reliability of the parts of a system, thus assuming that the system itself is free of design and workmanship flaws. From Section 4.2 we know that this is currently not the case for many CubeSats when they are launched, so relying purely on reliability prediction would be an incomplete approach. Nevertheless, as already pointed out, reliability prediction will be of interest for CubeSats as soon as more of them reach the constant failure rate region. Design decisions, commonly based on parameters such as power, volume, mass, price and availability could also be influenced by a reliability parameter. When doing that, it should be kept in mind that the specific characteristics of the space environment are mostly not considered in the available models. Thus, for example, failures emerging from high energy radiation cannot be fully predicted by the currently available models. We will discuss this further in Chapter 5.

For future missions, we evaluated current reliability prediction methodologies and approaches, and their suitability for CubeSat programs with respect to their mostly constrained resources. Since these methods are not necessarily developed for CubeSat projects, specific requirements had to be considered. Lack of time and lack of resources are critical aspects in CubeSat projects, so the approach should not impose excessive workload and enable a prediction with one week. Also, the method must be feasible without any experience in the field. Lastly, we wanted the method to have a quantitative output to be suitable for design trade-offs in the development process. Since we wanted to estimate the reliability early in the design process, the Parts Count method was chosen early as the suitable approach for our purposes. Parts Stress, Physics of Failure, FTA, FMEA and HALT (all described in Subsection 2.2.1) were ruled out for one or several of the aforementioned reasons. The Master's Thesis of Weisgerber [296], supervised by the author of this thesis, gives more details on this selection process.

Within the Parts Count method, different database offer reliability estimations on part level for a variety of selectable environments. Up until today, no data show clear superiority of one handbook over the other when dealing with CubeSats in space, we decided to use three different approaches for our estimation and compare the results. Of all handbook-based approaches, prediction by FIDES currently show the most promising results compared to on-orbit data (also see Subsection 2.2.1 and [139], [153], [154], [155]). The MIL-HDBK-217F, as the oldest but currently still most used approach, was also selected. Lastly, the IEC 62380 was also chosen due to its data being more up-to-date than the data of the MIL-HDBK-217F. The tool “Free MTBF Calculator” [297] was selected as the software for our estimations. It is available for free, has an easy-to-use user interface and allows the selection of different handbook-based approaches for components and parts on a variety of environments. The user interface is shown in Figure 4-147.

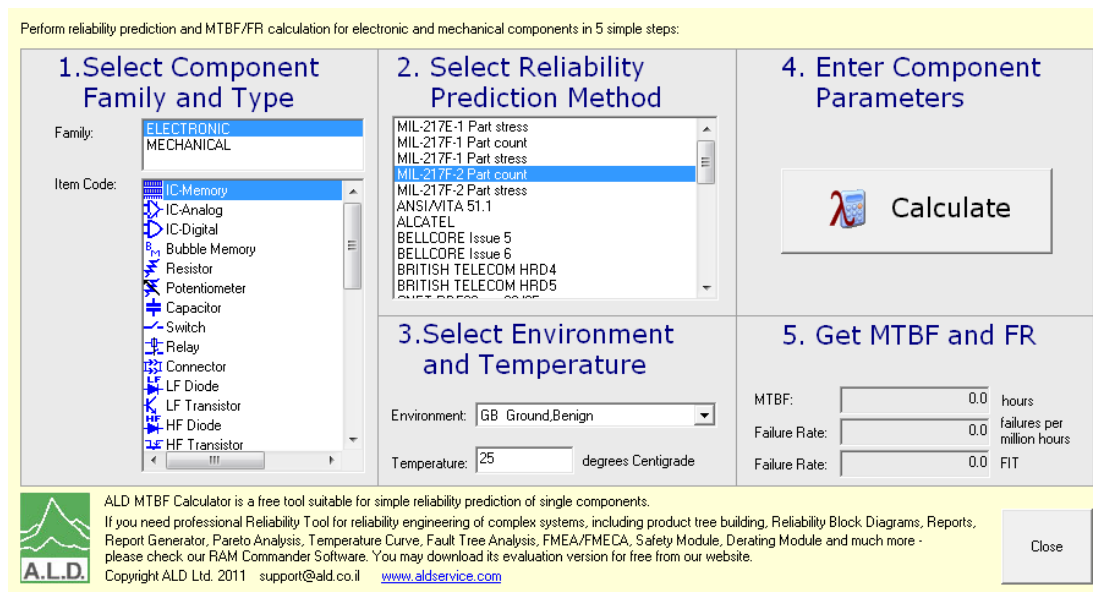


Figure 4-147: User Interface of the tool FREE MTBF calculator. Image Source: [296].

As described before, reliability assessment and growth modelling were in the focus of our efforts to ensure a reliable system, since our system, as many other CubeSats, still had many errors and flaws when we began system level testing. Thus, reliability prediction, working with a sum of the perfectly interacting parts, was never applied on system level. Nevertheless, as said before, the method can be helpful in the future as a valid parameter for design trade-offs, and that is also our goal to be presented in here. We selected our satellite’s Sidepanels as the target for our reliability prediction study on MOVE-II. The Sidepanel of MOVE-II (depicted in Figure 4-148) is a single-string system and as a simplification for our study we decided that any electronic part failure on it will lead to total failure of the panel. The solar cells and the surrounding circuitry was thereby not considered in this analysis. Also, the mechanical parts (hinges for the deployment of the Flappanel, attachment bolts – both not shown in Figure 4-148) were not considered for this prediction.

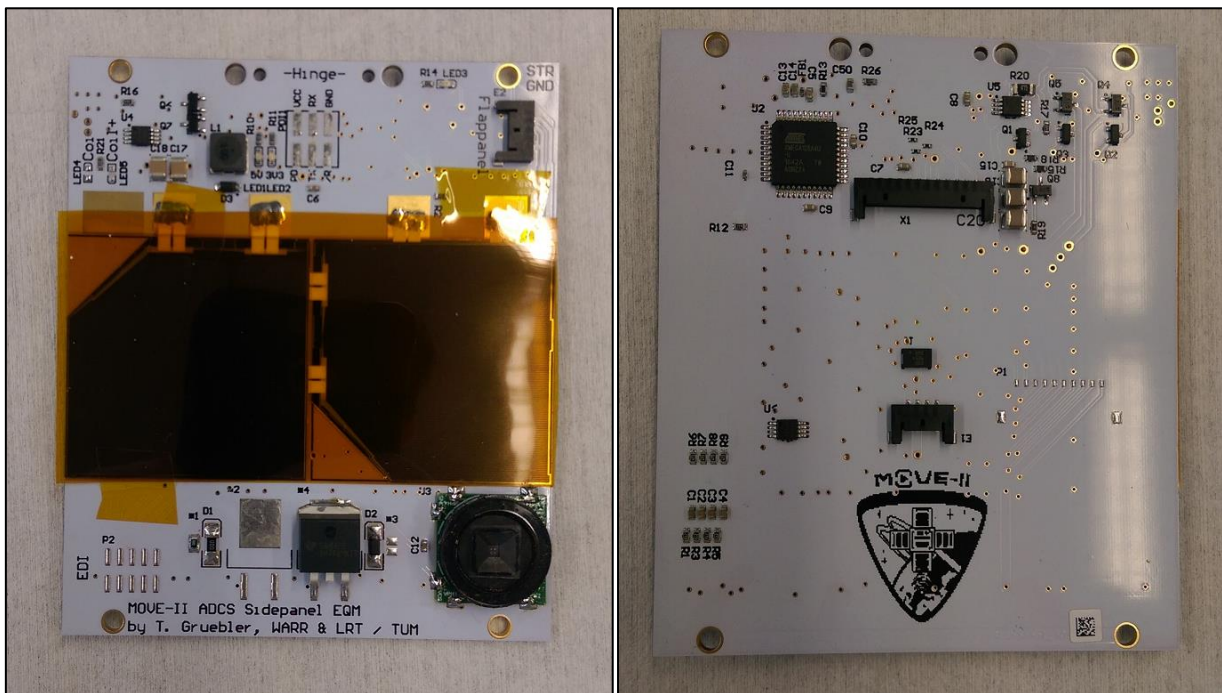


Figure 4-148: MOVE-II Sidepanel front- (left) and backside (right). The front-side faces toward outer space. Image Source: [296].

Thermal simulations of MOVE-II showed that the Sidepanels will reach a worst-case temperature of +6°C in the cold case, and a worst-case temperature of +20°C in the hot case. Thereby, the cold case is occurring when the satellite is leaving eclipse in safe mode, thus most of the subsystems switched off. Hot case means that the satellite is at full operation mode, actuating its Sidepanel integrated coils and is in sunlight but about to enter the eclipse. Since the Sidepanel showed similar temperatures in most test cases on the in- and outside, the aforementioned temperatures were assumed to be constant at both sides, simplifying the system further. For FIDES, we chose the application “Ground Mobile” for both temperatures in the Free MTBF calculator since no options for space usage were available. For MIL-HDBK-217F “Space Flight” was selected as primary choice for both temperatures, but also “Ground Mobile” selected as a reference to FIDES. For IEC 62380 “LEO permanent” was the best option available describing space use. This option was also complemented by “Ground Mobile” [296]. The results of all three estimations are typically given in so-called Failure in Time (FIT) rate. One FIT equals one failure per billion operating hours.

Table 4-3 summarizes the results of the reliability prediction of the Sidepanel. All part data were found using the Free MTBF calculator or other online sources. The analysis shows that depending on the handbook approach and the application chosen, the reliability of the Sidepanel is estimated between 125 FIT (t_{MTBF} of 912.6 years) and 45,655 FIT (t_{MTBF} of 2.5 years). 125 FIT resulted from the FIDES approach, using a ground mobile application (denoted as GM in Table 4-3) and a temperature of 6°C. 45,655 FIT stems from MIL-HDBK-217F, also ground mobile, at 20°C. As can be seen in Table 4-3, this estimation by the MIL-HDBK-217F is an outlier, which could originate from the harsh environment assumed for ground mobile applications for military purposes by this handbook. All other estimations are in between a magnitude, and range from the aforementioned t_{MTBF} of 912.6 years down to a t_{MTBF} of 86.3 years (MIL-HDBK-217F spaceflight application). The failure rate estimated by IEC 62380 is somehow in between, estimating 469 FIT (rounded) for a permanent application in LEO, which corresponds to a t_{MTBF} of around 243 years.

Table 4-3: Reliability Prediction for the MOVE-II Sidepanel with three different handbook-based approaches and the two temperature extremes identified. SF stands for spaceflight, GM for general mobile. Data Source: [296].

MOVE-II Sidepanel Designator	1 FIT = 1 failure per 10 ⁹ hours							
	Failure Rate for MIL-HDBK-217F-2 (in FIT)			Failure Rate for Fides 2009 Parts Count (in FIT)		Failure Rate for IEC62380 (in FIT)		
	6°C / SF	20°C / SF	20°C / GM	6°C / GM	20°C / GM	6°C / LEO perm	20°C / LEO perm	20°C / GM
*1	18.00	18.0	700.0	1.110	1.670	0.375	0.410	0.066
C1, C2, C3, C4	8.60	8.60	640.0	0.480	0.535	0.884	0.914	0.229
C11	8.60	8.60	640.0	0.480	0.535	0.884	0.914	0.229
C12, C14	8.60	8.60	640.0	0.480	0.535	0.884	0.914	0.229
C16, C19, C20	8.60	8.60	640.0	0.480	0.535	0.884	0.914	0.229
C17	8.60	8.60	640.0	0.480	0.535	0.884	0.914	0.229
C18	8.60	8.60	640.0	0.480	0.535	0.884	0.914	0.229
C5, C6, C7, C8, C9, C10, C13, C15	8.60	8.60	640.0	0.480	0.535	0.884	0.914	0.229
C50	8.60	8.60	640.0	0.480	0.535	0.884	0.914	0.229
D1	18.00	18.0	700.0	1.110	1.670	0.375	0.410	0.066
D2	9.90	9.90	269.5	0.872	3.279	14.019	14.113	11.287
D3	9.900	9.900	269.500	0.872	3.279	14.019	14.113	11.287
E1	44.000	44.000	900.000	1.940	5.595	12.892	16.214	5.983
E2	44.000	44.000	900.000	1.940	5.595	12.892	16.214	5.983
FB1	0.200	0.200	4.700	0.343	0.372	31.985	32.114	2.646
L1	0.200	0.200	4.700	0.343	0.372	2.731	2.853	0.777
Q1, Q2, Q3	37.950	37.950	880.000	0.638	2.593	13.937	13.956	11.305
Q4, Q5, Q6, Q7, Q8	37.950	37.950	880.000	0.638	2.593	13.937	13.956	11.305
R1, R3, R4, R5	18.000	18.000	700.000	1.110	1.670	0.375	0.410	0.066
R10	18.000	18.000	700.000	1.110	1.670	0.375	0.410	0.066
R11, R14, R26	18.000	18.000	700.000	1.110	1.670	0.375	0.410	0.066
R13	18.000	18.000	700.000	1.110	1.670	0.375	0.410	0.066
R15, R18	18.000	18.000	700.000	1.110	1.670	0.375	0.410	0.066
R19	18.000	18.000	700.000	1.110	1.670	0.375	0.410	0.066
R2, R12, R16, R17	18.000	18.000	700.000	1.110	1.670	0.375	0.410	0.066
R20	18.000	18.000	700.000	1.110	1.670	0.375	0.410	0.066
R21	18.000	18.000	700.000	1.110	1.670	0.375	0.410	0.066
R23, R24, R25	18.000	18.000	700.000	1.110	1.670	0.375	0.410	0.066
R6, R7, R8, R9	18.000	18.000	700.000	1.110	1.670	0.375	0.410	0.066
U1	14.247	14.247	106.858	6.220	20.330	44.606	61.317	11.872
U2	60.000	60.000	60.000	60.000	60.000	60.000	60.000	60.000
U3	31.350	31.350	715.000	1.373	3.812	17.583	20.489	11.322
X1	44.000	44.000	900.000	1.940	5.595	22.329	28.084	10.363
System	1322.047	1322.047	45655.258	125.116	196.734	425.990	468.890	262.517

Overall, it can be noted that the failure rates estimated by all handbook-based approaches clearly overestimate the reliability of the Sidepanel when looking at the results of past CubeSat flights. As already pointed out, this stems from the basic principle that the reliability is seen as the sum of the reliability of the Sidepanel's parts, neglecting design immaturities, workmanship errors but also failures that originate from software. As we have seen before, all three failure sources cannot be neglected when building a CubeSat, and software also must be considered when building larger satellites. Assuming 18 Sidepanel-like electronic

panels in MOVE-II⁸⁸, a total failure rate of in between 4.8 years (MIL-HDBK 217F) and 50.7 years (FIDES) results from the prediction. Although this is a great oversimplification of the process, which neglects the different electronic parts, temperatures involved and all mechanical parts of the satellite, it is believed by the author that such kind of reliability predictions will currently almost always overestimate the reliability of CubeSats in space. The reason for this is twofold: on the one hand, little on-orbit reliability data are known for most electronics used in CubeSats, since these missions mostly utilize modern COTS components that are often flown for the first time. On the other hand, many CubeSats implement new designs (as they are supposed to do) and thus to a large fraction face on-orbit problems stemming from those novel systems, such as design immaturities and also software errors, leading to infant mortality and DOA. Thus, all reliability estimations assuming that a CubeSat works perfectly and in the flat region of the bathtub curve are currently flawed.

Nevertheless, as soon as CubeSats have matured, such analyses could be useful to trade-off different design options. As can be seen in Table 4-3, part U2 is a large contributor to the un-reliability of the Sidepanel. U2 designates the microcontroller of the Sidepanel, which is an ATxmega128A4U-AU from Atmel. Within the early design process, the evaluation of other microcontroller or the implementation of a more complicated microprocessor solution could be influenced also by data coming from reliability prediction models. In our case, while neglecting the specific design of those solutions, the ATxmega could be replaced for example by the MSP430FR2433 from Texas Instruments, which has a predicted error rate of 5.6 FIT [298]. This would lead to a failure rate for the Sidepanel of 70.7 FIT at 6°C, which almost half of the predicted value of before (both estimated by FIDES). Clearly, for design tradeoffs, characteristics such as the footprint, power consumed, interfaces, heritage and many others must be considered, but the reliability prediction data could be one of those trade-off values, assuming that the development team is confident that most of the before presented reasons for infant mortality of the design are already eliminated or will be in the future.

It is the belief of the author of this thesis that CubeSats will mature and DOA and infant mortality cases will go down in the future. Despite that, the reliability prediction approaches presented in this subsection should not be seen as a replacement of the assessment methods presented in the earlier subsection. Thorough system level functional testing and environmental testing cannot be replaced by any prediction method that just assumes that a system works in a way it is supposed to do. Past systems that heavily relied on heritage showed that even that approach is not a guarantee that the system will work flawlessly. At the brink of mass-production of satellites, the presented assessment and prediction methods could also help those missions to achieve their goals. Assuming that the reliability of the first-of-its-kind model is assessed as presented before, the system passes its environmental tests, and the overall failure rate shows a saturation, subsequent models should show a quicker saturation, thus allowing a reduction of the functional and environmental testing time. Nevertheless, all models have to be qualified for launch loads and the space environment, since for example workmanship errors and failures in the material can occur independently of the heritage of the system. This could happen of course also in a reduced manner. Lastly, a reliability prediction using FIDES⁸⁹ could help on early design decisions in those mass-produced systems, especially since the selection process of parts not only involves one system but multiples of it, so the early decisions will have a greater monetary impact in the future. We will discuss these points further in the next chapter. The reliability prediction shown in this subsection is presented in more detail in the Master's Thesis of Weisgerber [296].

⁸⁸ Four Sidepanels and a stack of seven subsystems, which are assumed to carry the double amount of electronics (great oversimplification).

⁸⁹ Or any other predictions showing good accordance to on-orbit data and contain state-of-the art electronic components.

5 Discussion

“There is never a single right solution. There are always multiple wrong ones, though.”

– Akin’s 12th Law of Spacecraft Design

“Nullius in verba”

– Horace

(Also, the motto of the Royal Society)

We have learned in Subsection 2.1.3 that engineering flaws are often a reason for spacecraft failure, not the “classical” part failures assumed by today’s reliability prediction methods. Those failures emerge on subsystem or system level and are increasingly often originating from software than hardware. Software failures cannot be predicted and mostly also cannot be prevented by classical approaches (redundancy, etc.), as they are always design failures. The same engineering flaw in hard- or software can emerge in different areas on the spacecraft and as we have seen from the work by Leveson [61], diversity in programming teams can only partly mitigate that problem. The presented work on reliability assessments of CubeSats could be a first step to solve this problem, and this will be discussed in the following. We have also seen that COTS parts, especially when built for the automotive sector, do not pose many reliability disadvantages when used in spaceflight projects, and are suited well for space use as is (except for radiation and vacuum, which we will also discuss in this chapter). Examples from high performance commercial applications show that the state-of-the-art production process for high-volume electronic components achieves a higher confidence for the reliability of the parts than space-graded, low volume products. As an example, commercial hard disk drives, which are complex, high performance electro-mechanical systems, are by default qualified by running multiple 1000-unit test populations to mitigate infant mortality due to design issues [237]. Many performance requirements of these devices exceed typical spacecraft requirements, and on the established production line (when all infant mortality is flattened out) production units are also screened for workmanship/process failures through a 20 hour burn-in test by default [237]. This exceeds significantly the qualification process and the lot control that is achievable for space-graded products.

In small satellite and CubeSat projects, constrained resources pushed from the beginning towards the heavy reliance on COTS parts. Thus, classical random part errors should happen rarely in those kind of satellites, and this is supported by the feedback from on-orbit, as we have seen in Subsections 2.2.1 and 2.3.2 as well as in Section 4.2. On the other hand, many of these satellite missions are impaired by early failures or DOA, which originate from design and engineering issues, as the teams often work under very restricted resources, demanding timelines and often lack heritage and the experience of past missions. Traditional reliability prediction, summing up the part’s reliability figures for the specific application, will not help current CubeSat missions to achieve a reliable satellite. Also, since the quality of today’s electronic parts used in those missions is already very good, it doesn’t make sense to qualify on part level, as it would

be done in many larger missions. System level testing in a TLYF setup is in our opinion the key to improved reliability of CubeSats, and assessing these tests will help project managers in the judgement calls they will have to take.

As Hurley and Purdy [38] presented, true reliability is to measure how well a system performs in its operational environment. To simulate a complete space environment for the mission on-ground is a nearly impossible task, but in our view, the space environment is just one piece of the puzzle of “the operation environment”. Other pieces are for example: controlling the satellite over the complete communication chain (mission operations interface – ground station – antenna – satellite); getting confidence in the deployment system in different environments; testing the stability of software and algorithms in a HiL environment; and regularly carrying out 24h tests of the satellite. All those pieces and many more can be achieved on-ground, event by low-resources projects such as CubeSats. Hurley and Purdy [38] reported an example from a System Requirements and Design Review of a program, in which the reliability of a space system including launch was announced to be 90%. Clearly, these early predictions of reliability have nothing to do with reality and might cause wrong decisions to be taken within a space project. A reliable system, either for terrestrial or for space use, can only be assured by thorough testing it. As already pointed out, failures per year significantly decrease over time and a spacecraft surviving its first year on-orbit has an increased chance to survive its second year and the years thereafter too. Thus, design deficiencies and engineering flaws must be handled for CubeSats and small satellites before dealing with random failures at a constant rate or even wear-out. Nevertheless, it would be a mistake to solely rely on system level testing and skip well-established procedures of space system development such as reviews. The mindset of CubeSat developers should be to find as many failures as possible at any point in time in the system or subsystem, and not to prove that certain systems or subsystems “work as expected”. The before presented approach of early testing and careful characterization of complex subsystems, similar to the Bread-Brass-Silver-Gold approach of the Air Force Research Laboratory’s University Nanosatellite Program [286], [287] helps to reveal as many bugs and design flaws as early as possible. This is important, since the cost to correct problems will increase by an order of a magnitude for each step from concept over breadboard to the EM and later the FM, as for example Molnau & Oliveri [193] reported. We will discuss this further at the end of this chapter, also presenting recommendations for CubeSat developers based on our lessons learned (and the lessons learned of others).

Minderhoud & Fraser [299] showed that product development practices have evolved over the recent years. For many product categories it is now common to move from technology development over product development to volume production in no more than two to five years, a process that took previously 10-15 years. Product cost, time-to-market and quality have each become important characteristics for product development. We have seen in Chapter 2 that current satellites often take 10-15 years to develop and then have lifespans of 10-15 years. This lead to a “cannot fail” mindset and the Traditional Space Spiral (see Figure 2-44). In the future, reliability goals of commercial space missions might be also evaluated against the risk of on-orbit obsolescence, as soon as the faster technology cycles also fully arrive in the space domain, fueled by small satellites and CubeSats. Rivers [213] showed that small satellites and CubeSats could be the “bullets” in Collins and Hansen’s concept of “bullets versus cannonballs” of product development. In their concept, they showed that successful companies first tested a new market or approach with small experiments, so-called bullets, and if successful put their effort in a larger scale product in the same domain later on (cannonballs). Bullets, by their definition, must be low-cost, low-risk and low-distraction products, and thus especially CubeSats qualify to be bullets for the entire space market. An example for this is the US-company Planet, having released new versions of their satellites on a regular schedule before putting a bigger constellation for commercial purposes on-orbit. Another example would be precursor small satellite missions for Mega-Constellations such as OneWeb and the SpaceX constellation. Those constellations themselves utilize small satellites, thus departure from the long product cycles of the traditional space industry. CubeSats could also work as a bullet for the technology used on

those constellations. We will further discuss this in Section 5.2, and take a look at the risk of space debris caused by CubeSats.

5.1 Analysis of Satellite Reliability

As we have seen in Section 2.2, space systems are characterized by tight coupling and complex interactions. Subsection 2.1.3 and Section 4.1 showed us that 100% reliable spacecraft are not possible, nor are they feasible. As we have seen, there is an overall risk of 8.4% for any satellite to be lost due to a launch failure. Furthermore, work from Saleh & Marais [130] presented that infinitely reliable components do not exist – failure can be delayed, but will occur anyway – and that all improvements in reliability come at a price. They also showed that for a 15-year lifetime communication satellite, the highest net present value is achieved when relying on a 50-out-of-60 transponder approach, and not on a higher reliability.

We have seen that in general most errors in spacecraft can be broadly attributed to human error, as also Nieberding [69] pointed out. Tight coupling and complex interactions leave little room for errors, and often a series of complex, subtle events onsets a failure that was not understood or seen before. Also, as we have seen from the work from Leveson [60] [61] software plays an increasingly important role for spacecraft accidents. As she pointed out, the cause of an accident by definition can always be stated as a flaw in testing, since after the accident all conditions to cause the accident are determined and thus more knowledge is available than beforehand. As Paxton [300] showed, the greatest challenge for testing is to provide a full fidelity simulation of the system to onset failures. He also remarked, that failure review boards always have the ability to locate software and design issues, since they often know where to start looking. Nevertheless, both authors described in their work, mainly based on lessons learned of past missions, that end-to-end tests, thus TLYF, are the key to a successful test campaign. In our view, this must be complemented by changing our mindset when testing and dealing differently with failures in the future, both described in the following.

We already heard about the Faster-Better-Cheaper program of NASA in the chapters before. When the program was started in the early 1990's, people compared spaceflight in those days to the rise and success story of the commercial aviation industry. It took less than fifty years from the risky, first flight of the Wright Brother's to worldwide commercial aviation and they saw the same possibility for space [14]. However, one of the pillars that enables today's extremely low failure rates of commercial aviation is the dissemination of failures and lessons learned of any aircraft accident worldwide. For example, national institutions such as the German Federal Bureau of Aircraft Accident Investigation researches all errors involving aircraft within Germany and distributes thorough accident reports, including root cause analysis, to the public [301]. Furthermore, so-called bulletins of the spacecraft manufacturer are sent out to the airlines in case there are modifications or special maintenance needed. Of course, this stems largely from human life being at risk and from aircraft being products that undergo regular maintenance, which is both not the case for satellites. Nevertheless, little to nothing is done to disseminate lessons learned from success and failures of past missions on a regular schedule, neither from companies building large satellites [300] nor from small satellite and CubeSat developers. The rapid improvement of the airplane industry was only possible through sharing of lessons learned and experiences. The scarcity of satellite missions and data coming back from failed or partly failed missions must motivate us to talk more open about space failures and the root causes, despite the "can't fail" appearance some programs have to preserve for the public. This can't fail mentality might be resolved if we fly cheaper missions, and those missions more often, as it was already the idea of Faster-Better-Cheaper. We then should try to not put too complex and demanding goals on those small missions, and allow them to fail from time to time, while always having space debris mitigation as one of the basic goals of any mission.

The "can't fail" mindset impairs not only decisions in design, it also influences testing of the spacecraft and therefore the possibility to discover and resolve failures – the second mindset that has to change.

Spacecraft testing, and even more CubeSats testing, must be centered on the discovery of as many failures as possible. We tried to achieve this in our CubeSat by having challenges on who can provoke the most errors in the system on a weekly basis. It has to be kept in mind that this does not mean to destroy or harm the system, but to put it in as many situations as possible in a TLYF condition. This diversity in testing vectors, mostly achieved in a low-cost project such as ours with having as many different persons test the satellite as possible, is a key to successful low-cost testing. In projects with more resources, this could be replaced by automated random testing of software and the complete system. Nevertheless, it shall be kept in mind that humans, as they later also will command the satellite, might onset other failure cases than automated systems. Also, since our satellite has to work mostly on its own in space, we tried to often operate it only using the realistic, short communication windows, and left it on its own the rest of the time. Software errors such as run-out of memory or buffer-overflow might only occur, if the system is left on its own and flown in a space-like manner for a longer (> 2 weeks) period of time, using the complete chain of communication.

The analysis of the reliability of satellites in Section 4.1 showed that although a parametric fit matches the nonparametric data quite well, the time-dependent fractions of failed satellites have to be checked for consistency with their underlying Weibull functions. As we have seen, some functions presented by Castet & Saleh or Dubos et al. show an overall good alignment with the nonparametric data, but imply physical behavior that cannot be explained by the author of this thesis (late infant mortality, early wear-out). Mostly, this behavior stems from wrongly used scale-parameters within the different terms of the Weibull function. The failure rate of the mixed group of all satellites showed a right-open bathtub shape curve when fitting a modified Weibull function with a bathtub-like failure rate curve to it. This means that in the mixed group of 1,584 satellites no significant wear-out phenomenon was found. We already speculated that this might originates from masking of data due to the retirement of satellites. The second observation of the grouped data is, that it proves the earlier mentioned characteristics of spacecraft to either fail early or fail only with a relatively benign failure rate afterwards. In the studied group, the failure rate fell below 0.01 failures per day after one year. Thus, if a satellite survives the first year on-orbit, it has good chances to survive the second year too. Later used 2-Weibull mixture models showed that the reliability of the complete group is best described by a mixture of DOA, infant mortality, and failure within regular life (in our case modeled by a constant failure rate function). Due to the mixture of small, medium and large satellites, this parametric function might be incomplete, and no final answer can be given on the time-dependent failure rate of the complete group of satellites. Also, a root-cause analysis of the underlying nonparametric data might shift the parametric model in other directions. This analysis was not possible within this thesis, as the underlying data was not publicly available, and thus it was continued with analyses of the groups of small, medium and large satellites.

The analysis of small satellite reliability revealed that the parametric fits by Dubos et al. showed the same inconsistencies as the aforementioned fits of the mixed group of satellites. The reason for this was also the mishandling of the scale parameter of both terms of the 2-Weibull mixture function. The nonparametric reliability data model was rebuilt, and failures within the first nine days were merged to a subgroup of DOA cases. Although this might be an oversimplification and root-causes of the failures reveal other reasons, it is the belief of the author that a failure within the first nine days means that the spacecraft might never have been in a fully functional state in space. Failures within the first nine days account for more than three percent of reliability decline in the small satellite group, which saw an overall decline of 10%. The later used Single-Weibull fit showed similar results as the first used 2-Weibull mixture fit. As the introduction of additional parameters in a function is only meaningful if the fit itself is improved by this, the Single-Weibull fit is favored over the 2-Weibull mixture fit for the group of small satellites. Besides the DOA term, it comprises of a Weibull function with a shape factor of 1.2, thus describing a slightly increasing failure rate over time. Nevertheless, after a harsh decrease of more than three percent straight after launch, the reliability of small satellites decreases only with an almost constant rate of 0.5% per year. Thus, the already presented principle of surviving the early days somehow holds true, with the restriction that there will not

be an increased chance of surviving each following year after the second year for small satellites, but a slight decline. Without access to the detailed root-causes we can only speculate about the reasons for this behavior. The DOA rate could be caused by engineering flaws, incomplete testing and also failure of deployment systems. Small satellites often utilize deployment systems for their solar panels and antennas, as they are normally restricted to a certain limited envelope. As we have seen, those deployment systems benefit largely from heritage and it can be speculated that some small satellite developer may not have the experience or the heritage to reliably produce such systems. The near-constant failure rate decline over the following years could stem from multiple sources, and the results can also be altered by small satellites being retired before they fail. This could also be a reason why we cannot see a more dominated wear-out pattern in the data, although that could also originate from the fact that many small satellites rely on easier, more robust technology that might not wear out as fast as complex systems on larger satellites⁹⁰. High Energy Radiation might be an issue for small satellites, and in the case of SEE could be hidden somewhere in the data. TID effects on the other hand would mean that we see a certain wear-out pattern in the data, which we do not. Thus, we can speculate that TID might be not that critical for small spacecraft, since many COTS electronics nowadays work up to a TID of 10 or 20 krad without problems [302] [303], and the TID susceptibility of small satellites is furthermore benefitting from the relatively low orbits in which small satellites mostly operate. Nevertheless, it could also be argued that some satellites surely get more than 20 krad within the relatively long observation window. Although we cannot give a final answer on the reasons why the studied group of small satellites showed the failure behavior over time it did, it is certainly interesting to see the significance of DOA and the insignificance of wear-out for this group of satellites. Since the underlying nonparametric data stopped in 2008, it would be interesting to update the database and also filter the group regarding the experience of the developer. We will further elaborate on this in Section 6.3.

The class of medium satellites was the next group analyzed. Due to its scale factors, the fit presented by Dubos et al. showed infant mortality throughout the observation window, and wear-out that first sharply increased until $t = 7$ years, and then faded away until $t = 8.5$ years. After that point in time, medium satellites continued to fail in their model, but only due to the infant mortality term of the function. Clearly that cannot be explained by the normal characteristics of the infant mortality term of a 2-Weibull mixture function. We also fitted a 2-Weibull mixture function to the nonparametric data. Our function showed that infant mortality is of minor importance for medium satellites, as it leads to a reliability reduction of only around 1% within the first year (and fades away after). The second term of the Weibull function, described by a shape factor of 1.7, is of more importance, as it leads to a reliability reduction of 6% until the end of the observation window. Overall, the reliability of medium satellites is remarkably well within the observation window, and it can be agreed with Saleh & Castet [22] that they show a better performance than small or large satellites. Without knowledge of the underlying missions, it can be speculated that medium satellites are commonly produced by more experienced satellite developers, as the involved resources are usually larger. Also, there might be similar advantages in terms of wear-out as for small satellites. It could be the case that in the studied group, medium satellites fulfilled less-demanding missions than large-size satellites at thus might have less complexity and thus also have less risk for wear-out. Of course, this could also be caused by the owners of medium satellites retiring their spacecraft earlier than larger ones, thus preventing those satellites from wear-out. Lastly, the shape factor found for the second term of the 2-Weibull mixture function shows that there might be a mixture of constant failure rate and wear-out phenomena in this group. This answer can only be given by accessing the detailed root causes for every failed mission.

The original Dubos et al. fit of large satellites showed a better behavior of the wear-out fraction of the function. The infant mortality portion was similar to the small and medium satellites, as it continued to decrease the reliability of satellites until the end of the observation window. As two large satellites failed within the first week on-orbit, we included these two satellites as DOA in our 2-Weibull mixture function.

⁹⁰ An example for this are attitude control systems: while bigger satellites normally rely on wheels for their demanding missions, many small satellites can work purely with magnetorquers, which cannot wear-out.

Although the rate of DOA is quite small (reliability reduction of 0.5%), this shows that some large satellites might also suffer from deployment failures or other DOA failures. Interestingly, we found a constant failure rate term, with a relatively low scale factor of about one year, best fitting for the first term of our 2-Weibull mixture function. This term reduced the reliability by another three percent with the first three years on-orbit and faded out after that. The biggest portion of satellites failed due to the second term of the function, which was a Weibull function with a shape factor of about 2.6, thus describing wear-out. It reduced the reliability of the studied group by 11%, of which 8.5% stem from the last five years of the observation window. This strengthens the argument that large satellites might be the group of satellites most susceptible to the wear-out phenomenon. This could originate from several separate mechanisms. Firstly, large satellites in GEO might be operated until end-of-life, as the satellite's owner wants to maximize the return-on-investment. This point can be argued with since wear-out failures might prohibit the end-of-life disposal of satellites. Thus, responsible satellite owner should retire their satellite before it wears out. Secondly, large satellites might over-proportionally carry out high-demand missions that require complicated hardware, for example, three-axis pointing, mechanical gyros or other mechanisms, that can wear-out. Failure in one of those systems might be more critical than loss of one control axis in simpler missions. Overall, from the analyzed data, it cannot be concluded which mass class of satellites has a higher risk of failure than the others. At $t = 9$ years, the last observation point where reliable data for all three satellite classes are available, our parametric fit shows a reliability for small satellites of about 91.5%, for medium satellites one of about 93% and for large satellites a reliability of around 91%. This is within the prediction accuracy of all three models, so no conclusive answer can be given. The question is, if the different classes of satellites can be compared at all. The difference between manufacturing, mission profile, end-use, management, and many other crucial characteristics for satellite reliability seem too big to allow an objective comparison. On the other hand, coming back to our initial example of civil aerospace, the German Federal Bureau of Aircraft Accident Investigation analyzes all aircraft incidents, regardless of the size of the aircraft, and disseminates the information on those accidents afterwards. While this originally stems from the Chicago Convention on International Civil Aviation, it is also a holistic way of distributing lessons learned throughout the industry. A comparison of the reliability of a Boeing 747-400 with an Antonow An-2 might appear useless at first appearance but learning from other mistakes is not.

5.2 CubeSat Reliability

At the start of this work, publications by Swartwout already showed that CubeSats are more susceptible to infant mortality than other satellite classes. Swartwout also reported that many of these early failures could have been prevented by more exhaustive system level testing. Yet no collective time dependent behavior of the reliability of CubeSats was published when our efforts began. As we have shown, we collected the on-orbit reliability of all CubeSats launched until 06/30/2014 and censored the data on 12/31/2014. The data in our database mainly come from a survey, answered by 113 individuals affiliated with CubeSat projects in late 2014, publications and personal communication with many developers at conferences or via E-Mail. Although the CubeSat community is in the opinion of the author more open for a collective lessons learned process, it turned out to be difficult in many cases to track down what specifically happened with a CubeSat as soon as it was deployed in space. This partly originates from a lack of interest to share lessons learned, due to competitiveness, fear of funding cuts or other reasons. Partly it also stems from the lack of knowledge what exactly happened up there as soon as the CubeSat left the deployer. The latter point is especially the case for DOA and early infant mortality cases. Tools for diagnosis of spacecraft failures are telemetry, analysis of spacecraft operation, and retrospective analysis of failures when they are observed [54]. Often the latter option has to be chosen by CubeSat teams, since the other two need actual data from space, which are usually only in very restricted quantity or not at all available for failed CubeSat missions. CubeSat teams, especially those working within a university, often cannot rely on the experience and heritage of past missions most companies have, nor do they have an internal database in which lessons

learned are disseminated. Thus, sharing the lessons learned within the community would be of highest interest for that group of developers. Clearly more needs to be done in the future to collect and distribute the experience of already launched CubeSats.

The group of 178 CubeSats studied for this work shows clear signs of DOA and infant mortality. Thus, plenty of systems launched might have been insufficiently tested and thus were not in a stable operating condition before launch. Of course, the DOA and infant mortality cases could be caused by other reasons, as for example the loads during launch. However, the aforementioned work of Swartwout [249] shows that out of groups of CubeSats that underwent the same environmental qualification and acceptance tests, those which were built by less experienced groups failed to a much higher percentage than those which were built by experienced groups, despite all of the satellites passing the same environmental tests. More than 18% of all CubeSats studied by us were DOA, thus never achieved a functional state on-orbit. Fitting a Single-Weibull function to the nonparametric data, we also saw harsh infant mortality in the studied group, which reduced the reliability to below 60% within one year, and to below 55% within the observation window of 1.6 years. Thus, our data support the claim of Swartwout that CubeSats more frequently die early than other satellite classes. Since CubeSats always have to take more risk than larger satellites, as they are mostly experimentally spacecraft that fly novel designs for the first time, a generally higher rate of failure is acceptable in the view of the author of this thesis. However, it is also important that at least some data are received from most missions to be successful, and the DOA rate of more than 18% is clearly too high in that prospective.

The Single-Weibull fit was later complemented with a second term, describing a constant failure rate. This 2-Weibull mixture function changes not too much with regard to the too high DOA and infant mortality rate. While the DOA cases staying at the same level as described before, the infant mortality term of the 2-Weibull mixture function reduces the reliability of the studied group by 18%, and the constant failure rate term reduces the reliability by 6%. Clearly, the constant failure rate term is only of minor importance for CubeSat developers. We also showed in two sensitivity studies that the parametric fit does not change significantly when applying a cut-off at $t = 1$ year to the nonparametric data or when using the MLE method instead of nonlinear least squares fitting. Compared to the published results of Obiols Rabasa [254] (see Figure 2-48), the nonparametric data show similar results and deviates only by a few percentage points within our observation window. The nonparametric data for longer observation windows reported by him cannot be supported nor disproved by us, since the data available on CubeSats operating for longer periods of time becomes very scarce in our database. Also, many of the long-operating CubeSats stem from the early days of CubeSat development, when some satellites were mainly built as so-called Beep-Sats, thus without complex mission profiles and only with limited functionality. The parametric fit of Obiols Rabasa starts at a reliability value of one, and thus distinctively deviates from our parametric data. In the view of the author, DOA must be incorporated in the parametric data as well, as it describes one important characteristic of current CubeSat missions. If mission planners use the parametric data for estimating the overall historical chance of surviving their first month on-orbit or the numbers of spares needed for a CubeSat mission, they have to take the current rate of DOA for their predictions into account.

This current rate of DOA and infant mortality allows also speculations on the reason for the un-reliability of some of the current CubeSat missions. We have seen in the work of Birolini [39] that usually a system is assumed to be free of systematic defects and design failures at the beginning of life⁹¹. While this might be the case for most mass-produced products for terrestrial applications, it is sometimes not the case for space systems, and especially not the case for many CubeSat and small satellite missions. While 33% of our studied group failed due to an unknown reason, we can agree with Swartwout that system level testing would have prevented some of those missions and other missions with known failure root causes from failing. On the other hand, also large missions experience early faults in their systems, as we have seen for

⁹¹ This is also an important assumption needed for all reliability prediction methods.

example in past missions to Mars (see Figure 2-19). The difference however is, that large missions have the necessary backup-mechanisms (large ground stations, generous radio link, capability on the satellite to update and correct software) to correct anomalies after launch or the necessary redundancy to just use the system as is. Although the higher experience of the producers of the satellites plays a role when incorporating such functionality, small satellites and CubeSats always have to operate on far more restricted resources, and thus often don't have the communication channels, the functionality on the satellite, or the redundancy needed to cope with anomalies after launch. As said before, a certain fraction of CubeSats failing after launch should be acceptable for the developers, since they often try out novel systems that never have been flown before in space.

Nevertheless, there might be also a change in our mindsets needed, as the results to one of our questions of the survey showed. A difference of more than 20% between the estimated chance of failure for their own CubeSat with respect to a CubeSat built by others implies either that within the group of people we asked were many professional CubeSat developers that can be confident in their own skills, or that CubeSat developers in general might overestimate their own chance of success. In the view of the author, the results might be a mixture of both reasons.

Two points left for discussion is the influence of high energy radiation on the reliability of CubeSats and the impact of CubeSats on the space debris problem. While the former is generally not very well studied until yet, the latter is known quite well. In general, almost all electronics used in CubeSats are susceptible to the high energy radiation in Earth orbit. COTS devices as for example Flash memories can be impaired or even damaged by both TID and SEE [304]. The effects of high energy radiation always depend on a multitude of factors, ranging from the orbit of the spacecraft over shielding by structure or other electronics to, in the case of SEUs, internal error correction implemented by most terrestrial devices nowadays. An exhaustive analysis of this topic is therefore not possible within this work. Nevertheless, as already pointed out, it should be noted that most COTS electronics can resist low TID up to 10 or 20 krad as they are [302] [303]. Since current CubeSats overwhelmingly operate in LEO and have mission design lifetimes of one or two years, TID should have only minor effects on current missions. This of course will change in the future when more demanding missions fly in higher orbits (or on interplanetary trajectories) for a longer period of time. Hard SEEs such as SEL have to be mitigated by protection circuitry, which is nowadays sometimes already implemented by available COTS-subsystems for CubeSats. Soft SEEs such as SEUs might be corrected already to a certain degree using state-of-the-art electronics, which already come with error detection and correction capability to a certain degree. Also, automated watchdogs can be implemented easily on CubeSats to carry out autonomous reboots if parts of the system got stuck. Lastly, the use of modern electronics also allows a certain degree of redundancy in software on those systems, thus limiting the potential effects of bit flips by having multiple copies of the software on-board the satellite. Currently we cannot prove or disprove that a certain fraction of CubeSats already failed due to effects of high energy radiation. Nevertheless, the analysis of the cases for which the root cause is known suggests that only a very little fraction of all CubeSats might have failed due to radiation effects. As already pointed out, this could change in the future as soon as more CubeSats survive for a longer period of time and thus more data become available.

When talking about failures of spacecraft, we also must discuss the topic of space debris and the threat of an uncontrollable runaway effect, called Kessler-Syndrome, if the number of space debris increases with its historical pace in the future. Many discussions about CubeSats in the space industry revolve around the topic of space debris, and some of the traditional satellite developers denounce CubeSats as being just "a piece of space debris with an antenna" and as posing a great threat to the overall space industry due to growing launch numbers. In reality, de Selding [305] reported that one out of five CubeSats launched up to 2014 was not compliant to the international guidelines for space debris mitigation, meaning that those CubeSats will not decay within 25 years. Normally, the compliance of CubeSats, especially university-built ones, is supervised by National Space Agencies or other national entities, and a launch is only authorized

if compliant to the guidelines. So, the question arises why and how the non-compliant CubeSats were launched and for what purpose they had to exceed the 25 years on-orbit. From a reliability point of view, this clearly makes no sense since almost half of the CubeSat of the group we studied were already dead after the first year on-orbit. Since first-timers are performing exceptionally bad, as pointed out in the publications by Swartwout, the question also arises if it would be more feasible for those teams and for mankind to launch such first-time CubeSats only in very low orbits, for example using the deployment system of the International Space Station. A lifetime of a few months before decay seems sufficient for first-timers, as they mostly want to prove that their system will work in space. On the other hand, looking at the big picture, it seems a little bit unfair to portray CubeSats as “the problem” regarding space debris. Figure 5-1 by Swartwout [306] depicts all man-made objects on-orbit as of 2013. Clearly, CubeSats only contribute a tiny fraction to the overall problem of space debris so far.

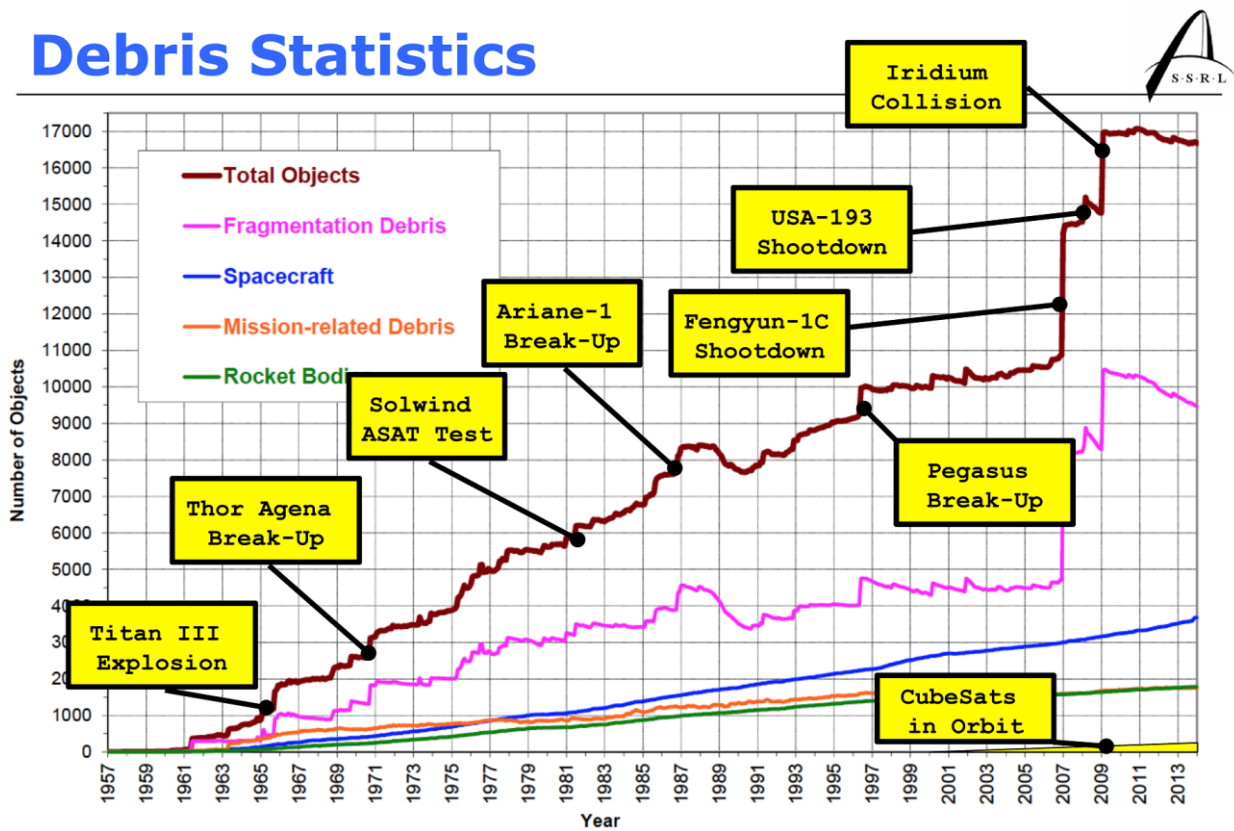


Figure 5-1: Number of man-made objects in space over time and different distinctive sources of the population. Image Source: [306]

Furthermore, recent work by the Inter-Agency Space Debris Coordination Committee (IADC) of the United Nations Committee on the Peaceful Uses of Outer Space (COPUOS) [307] shows that in between 2000-2015, 36% of all spacecraft in LEO were not compliant to the aforementioned 25 years deorbiting rule. This weak rate of compliance also extends to GEO, in which 60% of all satellites were not properly re-orbited in 2015 [307]. In that light, CubeSats, with their compliance rate of 80%, could be seen as role-models for the international guidelines.

Despite this, we are all sitting in the same boat, and predictions on the impact of a further increase of small satellite launches on the overall bad situation in some orbits [308] paint a grim picture of the future. Large constellations and swarms of CubeSats and small satellites, if not properly de-orbited in time, could cause a significant increase in on-orbit collisions, possibly leading to an uncontrollable runaway effect in the future. The risk posed by planned CubeSat missions is thereby dwarfed by the upcoming and planned Mega-Constellations, but despite this, space debris mitigation is a topic in which everyone must contribute to

preserve the overall environment. The future might bring more warranty issues when failed satellites are colliding with working satellites, as it already happened before. As CubeSat developers, we should carefully evaluate if we really need to exceed certain orbit heights for our missions and try to stick to the International Guidelines in any case. Otherwise there might be a time in the future in which it will become impossible to launch a CubeSat at all.

5.3 Reliability Assessment and Prediction of MOVE-II

As we have seen in Subsection 4.3.1, many lessons were learned while developing both MOVE satellites at TUM. University-built CubeSats enable hands-on education of students within a diverse project, but are often at the frontier of what is possible with such a small envelope in space. New hard- and software, but also new strategies how to develop satellites, can be tested on CubeSats in which failure can sometimes be an option. Thereby, the rise of the CubeSats is fueled by the rapid progress and miniaturization of commercial electronics. As an example, the data storage capacity of $9 \cdot 10^8$ bits⁹² of the interplanetary Galileo spacecraft [101], launched in 1989, is easily dwarfed by the storage capacity on MOVE-II.

To assure a reliable system, we showed that we focused on early prototyping and produced so-called brass-boards (all important functionality of the final system on not necessarily the final footprint) early, in order to carefully characterize and test our subsystems. We started system level testing in Flat-Sat configuration in early 2017, and originally planned with a least 9 months of system level testing. This was later extended due to launch delays. The two-model approach was in our case a worthwhile path to go, and we not only profited from many hours of system level testing and numerous failure corrections on the EM, but also on having two models available for parallel tests. The presented FRACAS and some examples how we tried to motivate our students to test as much as possible, were examples of our goal to keep testing and bug-reporting a straightforward and an easy thing to do. This helped us to achieve as many and as heterogenic testing vectors for our system as possible. The shown system level functional tests must always be complemented by environmental tests, and for the environmental tests it is best practice to stick to the experience of space agencies and launch providers. In any case, testing spacecraft always must start and end with the mindset of discovering as many bugs as possible in the system as long as it is on-ground – otherwise the errors will remain somewhere hidden in the spacecraft and might occur later while on-orbit. Further discussion and lessons learned on the general development process of CubeSats are presented in the subsection 5.4, summarized as “Recommendations for Building a CubeSat”.

Besides the importance of environmental tests in addition to functional testing, the reliability assessment presented is also strongly dependent on other factors. Firstly, a strong correlation can be assumed between time spent by people in the project/with testing and the occurrence of failures. In other words, if nobody tests the satellite, no failures will occur and thus the saturation curve might get compromised by that. Furthermore, failures might not be found when just operating the satellite in a standard way, especially at later stages of system level testing. So, both, average time spent by people in testing and heterogeneity of testing vectors are important parameters to consider when measuring the growth of failures over time. Unfortunately, we were only able to measure the first parameter as overall “time spent in MOVE-II”. This was tracked by students, voluntarily entering their working hours spent on MOVE-II into Redmine. Unfortunately, rather incomplete results appear when looking at those number, as can be seen in Figure 5-2. Firstly, the hours were only tracked until October 2017, and even before then, the student’s motivation of self-tracking already slowly faded away as the FM integration and acceptance tests were completed. Furthermore, this number is also influenced by a decrease of the number of students working in MOVE-II over 2017 (decrease from 100 to about 75 students). Lastly, although the reduced number of working hours could influence the overall result of the growth models, the decreasing number of other tasks in the project

⁹² Roughly 107 Megabyte.

over time must also be considered when looking at Figure 5-2. In early 2017, tasks such as the development of the mission operations interface or later the production of the EM and FM demanded time and resources. Later in time, it was possible to shift almost all efforts towards system level testing and error detection and correction. Reduced time or function for testing is not an isolated phenomenon of CubeSats. According to Hecht & Hecht [54], there are cases in which limited funding or interest also caused underreporting or under-detection of failures in larger satellites. This effect might also be observable when satellites are already in space and past their mission lifetimes. Thus, in any case, project managers should keep the motivation high through the testing process and the mission life for both detecting and recording anomalies and failures. For future applications it would be very interesting to track specifically the time spent in testing but also the time needed for failure correction, as it is needed for some reliability growth models. This is further elaborated in Section 6.3. Finally, testing and student motivation after December 2017 was heavily influenced by launch delays. Initially planned for December 2017, the launch of MOVE-II was shifted first to early 2018, later to April 2018 and currently (May 2018) it is foreseen for October 2018. This resulted in many students leaving the project and thus the testing time decreasing. Thus, although all bugs were tracked, and the saturation curve was modelled also thorough 2018, the data have only limited significance as less testing will always result in less bugs detected (i.e., a saturation of the curve).

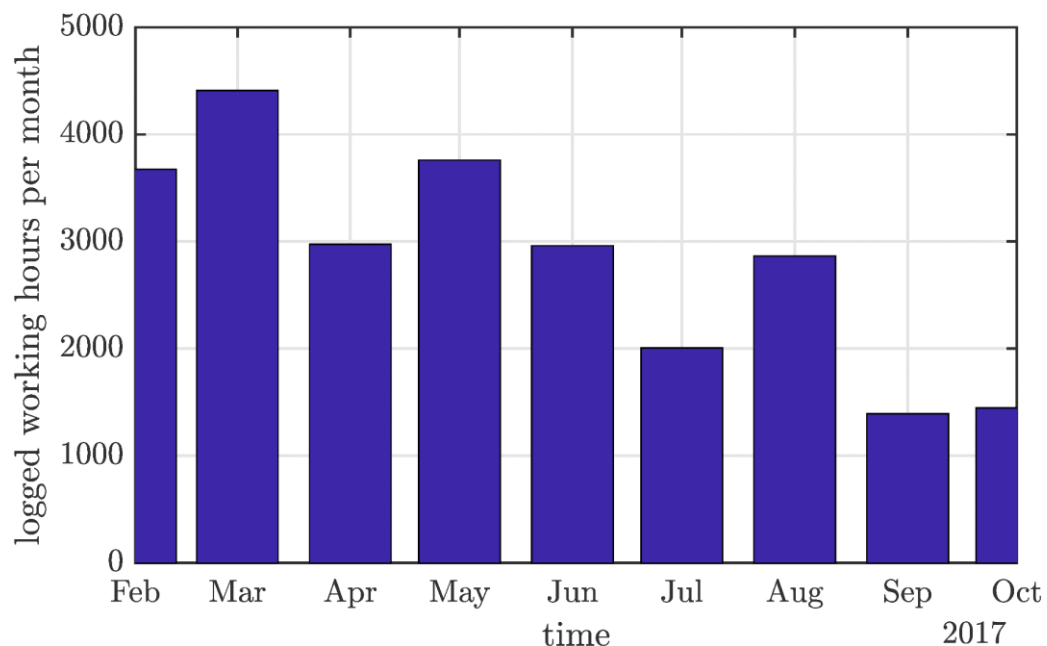


Figure 5-2: Working hours voluntarily spent & logged by students of the MOVE-II team. Image Source: [262].

Another important phenomenon, which we could observe while testing MOVE-II, was the masking of errors, mostly by other errors but sometimes also by limited heterogeneity of tests. This can be seen when looking at the saturation curves of certain subsystems, for example CDH, which shows a shifted start with respect to other subsystems. The reason for this phenomenon was mainly that the full functionality of some subsystems can only be explored if other subsystems work as intended, thus are bug-free to a certain degree. Until this is achieved, thus until the necessary subsystems reach a certain level of core-functionality, the occurrence of failures in other subsystems will be delayed. We also saw this phenomenon quite often within subsystems, mainly when correcting software errors. Having resolved an error and expecting the subsystem to work flawlessly, we often learned that only another error was waiting for us further down the road, which we were previously not able to detect, since it was blocked by the first error. Seldom, specific interaction was needed between subsystems to provoke a failure. This can also be seen as some kind of masking, but rather originating from the limited heterogeneity of testing vectors. As already pointed out, complex interactions and tight coupling make spacecraft vulnerable to these kinds of errors. Related to

masking, Hecht & Hecht [54] observed an on-orbit effect called “Shadowing”. By that, they meant the loss of observability for certain parts that are associated to components that already failed. The failed components might not be used anymore, thus failures in those associated parts might occur unnoticed.

Other points to be discussed consider the classification of errors (and their independence of each other), the quality of tests in relation to reliability growth, the influence of reduced pressure and thermal variations on the saturation, and if the system level tests and growth models might be a worst-case assumption of the later mission. We tried to classify errors (apart from the filtering into critical failures) and carefully analyzed each failure on its root-cause and if it is independent of other errors already known. Occasionally already existing errors were found again by other testers (as they were not corrected until then) and reported. Sometimes one bug emerged in different forms, and as aforementioned, we counted only the independent errors and discarded those that already were known. The quality of tests, speaking the heterogeneity, has massive influence on the overall results and the general confidence in the system one can gain. To test the satellite only in default modes would greatly simplify its later use and also provoke much less errors than when trying to explore the full state space of testing vectors. No airplane will be allowed to fly if only tested in “nice weather” conditions. This also influences the next point, which is validity of the growth model when testing the satellite mainly in not-space conditions. As already pointed out, the presented method should be complimentary to the established environmental tests and should mainly target design and engineering flaws that otherwise remain unnoticed. Nevertheless, the spacecraft must be exposed to the unique environment it has to sustain during launch and on-orbit. In the view of the author, already established environmental qualification and acceptance campaigns are sufficient to do this on system level, and might be complemented with selected environmental tests on subsystem level, as done in MOVE-II, to further reduce the risk of late failures and design changes. The growth models can and should be continued while doing those tests. As with other examples of concentrated functional testing, those tests might lead to a spike in failures when conducted for the first time.

In the view of the author it is not necessary to do all system level tests in simulated space environment in order to get valid results out of the growth modelling. As we have seen, many failures in current spacecraft (and also many failures found in MOVE-II) do not originate from the unique environment in space, but rather from design deficiencies and the general complexity of space missions. Thus, if the satellite is proper qualified for space use, heterogenous system level functional testing in lab-environment will rather complete the growth curve than invalidate it. Lastly, the presented growth curve and the reliability estimation based on the exponential model might be a worst-case assumption of the later usage. It could be argued that some of the functions tested or simulated will never occur in space, thus certain failures discovered never would have happened. On the other hand, this point could be reversed, as many past missions have shown that certain situations occurring in space were never imagined beforehand. Of course, it would be presumptuous to assume that we covered exactly the right tests to prevent this from happening in space, but heterogeneous testing vectors are our way of at least decreasing the chance of that. The presented estimation of reliability and number of bugs left in the system might be a worst-case assumption, but without a statistical relevant number of satellites in space any sound estimation is hard to achieve. Past data (see Figure 2-13 right) show that for some early NASA missions there was a spike in the failure rate on the first day of life, which diverted from the estimations by the then used growth models. This divergence disappeared after a few days on-orbit, and later failure rates showed a good alignment with the growth models. Thus, the estimations by our models might be compromised by a yet unknown phenomenon that reduces the reliability of spacecraft significantly within the first days. Of course, this then seen reduction could also originate from the aforementioned main reasons for DOA and infant mortality, and thus might be prevented by the functional testing we propose and already incorporated in our growth models. Lastly, a worst-case prediction of the total number of failures in the system and the reliability on-orbit might not be satisfying, but in the view of the author it is better than predicting too optimistic results and skipping functional testing, either by having no prediction models at all or wrongly using constant failure rate approaches when the reliability of a new-designed system is not proven by tests.

Coming back to the results of the reliability growth models, Table 5-1 shows the number of errors estimated by the five models at different points in time. As already pointed out, the basic exponential reliability growth model and the exponential reliability growth model with variable starting date showed the best stability when using a reduced dataset. Thereby, the basic exponential model shows a slight overestimation of total errors when going back in time to $t = 219$ days, but delivers relatively stable results afterwards. The exponential reliability growth model with variable starting date shows stable results over the full period, with a total estimation range of 17.4 errors. Overall, both models predict that there were still errors left in the system at the end of the observation (12/23/2017). All other models had weaknesses when used with our limited data. The Yamada & Osaki model, though predicting the overall failures as accurately as the basic exponential model, showed a too big dispersion of all parameters since no classification in easy- or difficult-to-detect errors was possible within MOVE-II. As aforementioned, this classification task might be too difficult for CubeSat teams, since a lot of experience is needed to classify errors accordingly. The two S-shaped models were discarded since they showed constantly increasing estimations of errors with increasing errors found. While little increase could be tolerated, constant increase means that the models are not of use for any prediction within the development time, since those predictions have to be presumably increased at any time in the future. In case of the delayed S-Shape model, the situation is further worsened since the model under-predicts the amount of errors in three of four occasions. As already presented in Subsection 4.3.2, there could be multiple reasons for that, ranging from the delay between detection of error to correction to a constant error detection rate needed in case of the delayed S-Shaped model. Both models assume a mutual dependence between faults, and this is also a pattern we have seen in MOVE-II, although the percentage of this dependence was not studied. Contrary to the S-Shaped models, the exponential models assume a mutual independence of the errors [179], which was also true for errors seen in MOVE-II. Clearly, more work is needed on this topic in the future and more data from other CubeSat developments would help to get confidence in one or multiple of the presented models.

Table 5-1: Total number of failures predicted by the five different models, and their estimation of hidden failures at t_{max} and the absolute width of estimation. Green fields depict an estimated number of errors greater than the already number of errors found, yellow estimations that are below the number of errors found.

	Timerange				Estimation Hidden Failures at t_{max}	Estimation Width [Failures]
	$t_{max} - 100$ days	$t_{max} - 40$ days	$t_{max} - 10$ days	t_{max}		
Known errors at time	363	402	427	432	-	-
Basic exponential reliability growth model	561.1	500.7	505.1	507.1	75.1	60.4
Yamada & Osaki reliability growth model	553.2	498.8	504.1	507.2	75.2	54.4
Exponential reliability growth model with variable starting date	477.6	460.2	472.9	477.1	45.1	17.4
Delayed S-shaped software reliability growth model	368.4	389.2	403.4	408	-24	39.6
Inflection S-shaped software reliability growth model	406.4	417.2	444.5	453.9	21.9	47.5

Finally, we can compare our results with an earlier point in time, since the models were first used in the Master's Thesis of Schummer [262], supervised by the author of this thesis. His results of the reliability estimation also showed the overall better stability of the basic exponential reliability growth model and the exponential reliability growth model with variable starting date. Remarkably, already on day 168, the exponential reliability growth model with variable starting date predicted a total amount of 450 errors in the system, and changed this estimation only within a range of 27 errors since then. The total amount of errors found during that time was 331. The basic exponential model on the other hand, starts to overestimate the total number of errors at a certain point back in time. At $t = 168$ days, it foresees 614 errors in the system,

but decreases that estimation over time. This not ideal behavior of the model is perceived as more favorable than a constant underestimation or an increasing prediction rate, as also seen with both S-Shaped models in the data of Schummer. Future improvements in detection and classification might help to make the Yamada & Osaki model usable for CubeSat teams, but until then this model is also of inferior quality.

As presented in Subsection 4.3.2, the growth curve of critical failures was studied in order to estimate the number of potentially fatal errors left in the system. This was later used for the before discussed reliability estimation of the system. Table 5-2 shows the stability of all models when fed only with the data of critical errors. As before, and for the aforementioned reasons, the basic exponential reliability growth model and the exponential reliability growth model with variable starting date deliver the best results for the group of critical failures. The estimation of the basic exponential model has a width of 2.6 critical errors in between the estimation at $t = 219$ days and $t = 319$ days. The offset between the estimation and the number of critical errors found is 3.8. The model with variable starting date shows a similar variation of 2.9 critical errors over time, and estimated a total number of 112.9 critical errors, which remarkably close to the 113 critical errors found in MOVE-II. Combined with the before presented total number of errors, the model with variable starting date seems to deliver the best results when fed with data from different points in time.

Table 5-2: Total number of critical failures predicted by the five different models, and their estimation of hidden failures at t_{max} and the absolute width of estimation. Green fields depict an estimated number of errors greater than the already number of errors found, yellow estimations that are below the number of errors found.

	Timerange				Estimation Offset [Failures]	Estimation Width [Failures]
	$t_{max} - 100$ days	$t_{max} - 40$ days	$t_{max} - 10$ days	t_{max}		
Known errors at time	105	112	113	113	-	-
Basic exponential reliability growth model	119.4	117.2	117	116.8	3.8	2.6
Yamada & Osaki reliability growth model	350	1174	1013	972.2	859.2	824
Exponential reliability growth model with variable starting date	110	111.8	112.7	112.9	-0.1	2.9
Delayed S-shaped software reliability growth model	101.5	105.4	107	107.4	-5.6	5.9
Inflection S-shaped software reliability growth model	100	107.4	109.8	110.3	-2.7	10.3

This slight advantage of the model with the variable starting date is also supported by a final sensitive analysis, which is shown in Table 5-3. The model with the variable starting date is the only model capable of delivering an acceptable width of estimation and is simultaneously not under-estimating the total number of critical failures in the system. It delivers relatively stable results of the total number of critical errors since approximately half of the total time spent in system level testing so far. The results of this further sensitivity analysis are also presented in Figure 6-16 in Appendix B.

To summarize, the presented reliability assessment methods and growth models could be a first step towards CubeSats with less DOA and infant mortality rates. To support the results found in this thesis, the flight data of MOVE-II and more data from other CubeSats are needed. In the following last subsection of the discussion, we will focus on recommendations for building a CubeSat within a university environment. Although it might seem to be a random aggregation of useful advice and lessons learned, it is loosely structured around the different phases of CubeSat development, and also covers project management advice at the end of the subsection.

Table 5-3: Total number of critical failures predicted by the five different models for earlier points in time, and their estimation of hidden failures at t_{max} and the absolute width of estimation. Green fields depict an estimated number of errors greater than the already number of errors found, yellow estimations that are below the number of errors found.

	Timerange				Estimation Offset [Failures]	Estimation Width [Failures]
	$t_{max} - 200$ days	$t_{max} - 150$ days	$t_{max} - 100$ days	t_{max}		
Known errors at time	87	99	105	113	-	-
Basic exponential reliability growth model	226.5	123	119.4	116.8	3.8	109.7
Yamada & Osaki reliability growth model	350	385.5	350	972.2	859.2	622.2
Exponential reliability growth model with variable starting date	128.1	106.4	110	112.9	-0.1	21.7
Delayed S-shaped software reliability growth model	98.8	96.2	101.5	107.4	-5.6	11.2
Inflection S-shaped software reliability growth model	81.7	89.4	100	110.3	-2.7	28.6

5.4 Recommendations for Building a CubeSat within a University

This subsection is partly based on two conference papers ([244], [275]) by the author of this thesis.

Coming back to the potential enhanced role of small satellites and CubeSats in the future, we try to summarize the key lessons learned of the development of both of our CubeSats as well as other small satellites in this subsection. Also, as already pointed out, the Faster-Better-Cheaper program of NASA still holds many lessons for those who try to reverse the Traditional Space Spiral. Thus, we will also incorporate lesson learned of this program, mainly based on the book by McCurdy [14].

Beginning with the development process itself, the importance of building and testing hard- and software early and often was already mentioned in Subsection 4.3.1. Through the Bread-Brass-Silver-Gold Approach of AFRL [286], [287] it can be assured that many design flaws are captured early in the process, and demonstration of core-functionality of subsystems is also achieved early. For us, this was also a main lesson learned from First-MOVE, and implemented in MOVE-II. The careful characterization of complex subsystems first through individual functional and performance tests under a variety of relevant environments, followed by a flat-sat style and increasingly integrated demonstrations and characterizations, must happen early on and with large scrutiny. Especially temperature-dependency of electronic circuits, from frequency, voltage and current shifts to change in timing constants must be understood and tested ideally in thermal vacuum environments or at least at the temperature extremes. To track all testing efforts, our subsystems (and in the case of more complex products sometimes also parts) were QR-coded by us. This allowed an easy traceability of every subsystem, and made life easier for the testers of MOVE-II, since they only had to scan the specific QR-code to fill in specific subsystem/part test data. The aforementioned online accessibility of the satellites (also of the HiL test-benches), complemented by selected approaches to enhance the motivation for testing, played a major role in the extension of system level testing and the diversification of testing vectors. In all development efforts, testing and testability should be considered as an important characteristic of a good design, as explained in the section on our shape memory alloy HDRM. Design freezes and deadlines for major parts of the hard- and software being finished and demonstrated have been very useful in MOVE-II to both achieve the aforementioned early testing of all subsystems but also to keep the motivation and dedication of the team at a high level. Whenever possible during development, having the TLYF approach in mind helps to incorporate necessary interfaces or debug outputs already in the design. This mindset also stimulates to design, build and test GSE, GS and the mission operations interface needed for that approach early on. A lesson learned of MOVE-II will be that

testing of the mission operations interface started too late. Finally, a lot of discussions are still ongoing if COTS is suitable for space use or not. As we have seen in Subsection 2.2.3, mass produced COTS components for the automotive sector fulfill or exceed many requirements needed for space usage, except for vacuum and high energy radiation. As already pointed out, the production lot size of those terrestrial components dwarfs those of specific space produced components, and thus the part quality can be assured to a higher confidence for the terrestrial ones. Though the behavior of COTS parts in vacuum must be tested beforehand, there is no general show-stopper for their applicability to space environment. In terms of high energy radiation, modern electronics often have better built-in error correction, which is already sometimes capable of self-correcting soft SEE's. Hard SEE's on the other hand must be covered by appropriate design measures, as almost all COTS parts will have some susceptibility to them. The tolerance of COTS with respect to TID effects depends on the specific orbit and mission time. A radiation dose of up to 10 krad can be tolerated by most COTS parts, and this is sufficient for most short-term CubeSat missions in LEO. For higher orbits and longer mission lifetimes, shielding could be applied to protect the most vulnerable circuitry from TID effects. This should be carefully selected though, as shielding can always cause unwanted secondary particles and sometimes enhance the vulnerability of circuitry behind it, if not well designed.

Software development and the associated lessons learned could fill an own chapter, and both the increasingly importance of software for space missions and the rising number of space failures stemming from software should underscore its importance. As we have seen, software errors are always design errors, and thus redundancies or other traditional mitigation strategies will not work for them. The high complexity and tight coupling of spacecraft emerges today often as software failures, and "complete" testing is rather hard to achieve for software. Moreover, failed space missions show that the reuse, optimization or other adaption of heritage software might also end in chaos. Thus, also in CubeSat projects, more time should be spent in careful design and early testing of software, and that is definitely a lesson learned of MOVE-II. Modern terrestrial software design and testing approaches allow, if carefully prepared, automated testing of software parts early on. Also, mission managers and group leaders have to make decisions early on not only on the hardware but also on the software part of the spacecraft, in order to facilitate the early testing of software similar to the process in hardware. While this might be already done in other projects, this is a point we have to improve after MOVE-II. Finally, too often changes and late modifications are implemented in software, and often those changes are used to cope with errors and flaws of other origin than software. Considering the history of space missions that failed due to software flaws, all involved developers have to understand that late software changes might be as critical as late hardware modifications. No reasonable person would design and produce a new CubeSat structure one month before launch. Yet for software these modifications are more tolerated for some reason across the community. Although software updates often saved functionality and prevented other failures as soon as the spacecraft was on-orbit in the past, design freezes also should be enforced more rigorously on software too. Although we tried to implement that in MOVE-II through a GitLab based approach, in which at least four people and the project manager had to accept software modifications before they were put in the Master Branch, we also learned lessons to improve that in the future.

Although classical mission assurance guidelines do not foresee a broad review process for Class D⁹³ missions [242], this process and the inclusion of external experts is in our view an essential keystone of successful CubeSat development in universities. We learned this lesson from First-MOVE, in which several tests and (sub-) reviews were initially not deemed necessary because the satellite was "only" a CubeSat. One important lesson learned was that a CubeSat is only slightly less complex than a larger satellite, thus the development project needs more or less the same number of technical reviews, despite the small size if the spacecraft. Both the Preliminary Design Review (PDR) and the Critical Design Review (CDR) had major impacts on MOVE-II, and many design deficiencies were revealed in both meetings that would have either

⁹³ Remember that a Class D mission is typically in between US\$15 million and US\$250 million.

remained unnoticed or would have occurred much later, presumably while testing the spacecraft or on-orbit. More than 40 external experts were involved as volunteers in the PDR and CDR of the satellite. We believe that expert knowledge of senior engineers enhances both hands-on education of students in different subsystems as well as technical quality, preventing pitfalls during the design phase of the satellite. The reviews are not only important as a gate for technical maturity, but also vital for having short-term goals for the student team. Although there are some management resources needed for the identification and coordination of those voluntary experts, the possible benefits of expert knowledge transfer from experienced professionals outweigh the costs. Although we achieved some improvement in MOVE-II regarding the involvement of external experts, there are definitely lessons learned again from this project that we will try to implement in follow-on missions.

Another important lesson learned in MOVE-II and a general lesson from small satellite development concerns the complexity of small spacecraft. In general, the reliability of small satellites and CubeSats could profit from their reduced number of parts and reduced complexity, as shown by Wertz [30]. On the other hand, an (often unnecessary) increase in complexity can be observed for many CubeSat missions, and this was also observed during the Faster-Better-Cheaper program of NASA, as we have seen. This complexity growth stems mainly from two sources, both shown by McCurdy [14]: Firstly, most spaceflight engineers try to mitigate errors by adding more safety features and redundancies in the system, which itself raises system complexity and thus also increases the chance of failure. Secondly, the frequency of satellite launches for institutions also defines the complexity of the spacecraft, since seldom launches create a situation in which developers and scientists want to put as much as possible on a single mission. The latter point can also be observed in some CubeSat teams, as rare opportunities also lead to complex and demanding missions, contradicting the original intent of CubeSats. For MOVE-II, we observed both, and overall the satellite is far too complex for the goals it should achieve. On the other hand, an educational environment should always accept new designs of students to some degree, as that is one of the main goals of those programs. Yet the general complexity of CubeSat missions should be kept low, and complexity should not be mistaken with sophistication, as Ward [215] showed. The same was observed during the Faster-Better-Cheaper missions, in which the relationship between cost, schedule and complexity was impaired by the failed missions. As McCurdy [14] analyzed in his book, the failed missions of Faster-Better-Cheaper reduced costs and schedule faster than complexity. Or in other words, looking back at Figure 2-42, they had too much complexity included in their missions with respect to their resources and schedule. This does not mean that those missions failed did not test their spacecraft well enough with respect to the resources and time they had. The enhanced complexity, as for example in the Mars Polar Lander mission, required the interaction of three separate systems to provoke failure [14]⁹⁴, and this is the inherent problem of adding more complexity to space missions. High complexity and tight coupling, as already pointed out, are much harder to test and also weaknesses of the design are much harder to identify beforehand [14]. Thus, if high complexity is needed, which in case of both aforementioned Mars missions was the case, it must be covered with adequate resources and time, and must resemble in even more testing efforts on system level to uncover failures that stem from this high complexity of the system. On the other hand, most CubeSats will successfully fulfill their mission goals without introducing too much complexity in their systems, and they might even better achieve their goals if they are not too complex. This is also one of the main lessons learned of the successful Faster-Better-Cheaper missions. As McCurdy [14] reported, half of the “good ideas” were rejected during the design phase of the later successful NEAR mission, simply since they would have increased cost, schedule and complexity too much. Cost and schedule control are perceived as minor goals in larger space missions, and highly sophisticated missions sometimes rightfully demanded both to be increased for the high complexity needed to fulfill their missions. On the other hand, the Traditional Space Spiral (shown in Figure 2-44) will always lead to fewer missions,

⁹⁴ This characteristic (several separate systems needed to provoke failure) could also be seen as one reason for the failure of the Schiaparelli Lander of ESA in 2016.

increasing their complexity due to scarcity, and thus drive costs and schedule up. With small satellites and CubeSats we can reverse that circle and become a vital and important part of the space community, but only if we keep the complexity of our missions low. If not, we will also end up in the no-fly zone (see Figure 2-42).

Testing the CubeSat and its subsystems deserves a separate paragraph in this condensed subsection, since designing and building a CubeSat with students will always involve novel ideas and approaches but also immature designs and engineering flaws that have to be detected at latest through testing before the spacecraft is launched. In that point we learned many lessons from First-MOVE, although the environmental test campaign of the satellite was carried out well, and the satellite passed all tests. Despite this, a lesson learned from First-MOVE was to further extend in-house subsystem and integrated system-level testing of all components, including the purchased subsystems. As already pointed out, thermal cycling tests were significant for the satellite, since a major, temperature-based issue on the latch-up protection unit was only found during those tests. Overall, since CubeSats are highly integrated systems, the careful planning of testability, integration and accessibility of all subsystems cannot be underestimated. In MOVE-II, we tried to solve that by designing and by producing additive manufactured prototypes early on, as shown in Subsection 4.3.1. Also, the necessity for longer, continuous operation tests of the fully integrated system is a lesson learned not only from the First-MOVE team but also from other CubeSat teams worldwide. We learned that in hindsight, the overall testing time of both the major sub-systems and especially of the fully integrated system was insufficient to ensure reliable and successful operations for First-MOVE. In MOVE-II, we tried to maximize both the time spent in system level testing and the diversity of testing vectors. Testing should be as easy as possible for the students involved (i.e., satellite and testing equipment is controllable online), otherwise it will not occur frequently. If decisions are needed what to test best with limited time, efforts are best spent on known weak links [237] in order to increase the chance of mission success. For CubeSats, this especially means tests of deployments systems, if implemented, and the subsequent first 24 hours of mission life, preferably in a TLYF approach with the full command chain including control software on-ground. Also, thermal-vacuum testing and testing of launch loads (vibrational loads, acceleration loads) are in that terms weak spots that must be tested. Furthermore, automated testing of sub-assemblies and software could reduce the overhead needed for thorough testing. Tosney & Pavlica [37] presented additional general best practices, of which some are applicable to CubeSat Development as well: Early interface and harness checks on all hard- and software can be done if the project follows the aforementioned development approach of early prototyping and early hardware production. The test of high power electronics and RF hardware in TV prior to system level test is supported by the mentioned approach to already test selected subsystems in TV, in order to understand their behavior in this environment. Also, for MOVE-II, system level tests with active RF communication were carried out in TV. Rigorous system level testing will also help to identify problems originating from electromagnetic interference (EMI) and electromagnetic compatibility (EMC). Tosney & Pavlica [37] also recommended that a realistic schedule projection of testing is needed and a disciplined anomaly tracking and resolution shall be implemented. We tried to implement both in MOVE-II using the methods presented in Subsection 4.3.2, but also see possible improvements for future projects. Finally, the importance of following a rigorous TLYF approach cannot be overstated. Although TLYF is not formally defined in existing government standards or handbook [37], it means to test the satellite (or parts of it) in conditions as close to the flight as possible. We learned in First-MOVE that this not only involves the satellite itself, but all necessary infrastructure on the ground as well. In MOVE-II, as already pointed out, we try to use the full chain of command as standard way of testing the satellite, deviating only from this in justified exceptions.

On the management side, the university environment dominates many decisions program managers have to make and also obstacles they have to overcome. Other than in professional organizations, CubeSats in a university environment include a) planning the project around students' academic schedules rather than in a traditional, linear fashion, and b) the careful selection and assignment of team members to subsystem teams in order to retain student motivation and an even distribution of more and less experienced members.

A key lesson learned in First-MOVE concerned the drain of knowledge about the satellite or its subsystems due to students leaving the project, having either completed their thesis work or their respective graduate or undergraduate programs. To remedy this situation, which many university-based CubeSat teams experience, interested students should not just be assigned to a specific thesis topic, but also encouraged to stay involved in the project before the beginning and beyond the duration of their thesis work. Since retention of specific students over longer periods of time is very difficult in the German academic curriculum, favoring diverse projects over specialization, we had to choose a different approach for MOVE-II. We used a voluntary, dedication-based approach through membership in a student association and thus improved both academic success and retention of knowledge. All involved students are grouped in a LRT-associated student organization (Scientific Workgroup for Spaceflight and Rocketry, WARR, [309]) in which their commitment to the project is out of interest and dedication, rather than short-term academic credit, leading to less team fluctuations. This also involves a need for enhanced educational and academic outreach to recruit and train new students. Even if these measures are implemented, launch delays are always obstacles that CubeSat teams have to face and in which massive knowledge drain is almost inevitable. First-MOVE faced multiple launch delays in between 2009 and 2013. This extensive time delay caused a significant knowledge-drain due to the fluctuation and graduation of the involved students and few longer-term staff members. This issue could only be partially addressed with written documentation through academic and project-relevant documentation. MOVE-II also experienced launch delays to a lesser degree so far, with its launch initially planned in December 2017. We again learned some lessons from this, as the hiring process of new students stopped in summer 2017 due to the then nearing launch date, but was re-established in mid-2018 as new students were needed to replace those who graduated in the meanwhile.

Despite these rather unique obstacles for the management of university-built CubeSats, many lessons can be also learned from small satellite projects, and those are presented loosely in the following. Overall, strong management with minimum layers of authority, team continuity and small dynamic teams with an engineering culture are the most important contributors to mission success for Class C/D missions, according to a report by the Aerospace Corporation [238]. Also, besides mission complexity, management mistakes were one of the reasons found for failure in the Faster-Better-Cheaper missions. McCurdy [14] pointed out that many of the project managers failed to follow the teamwork principles needed for the small teams of the Faster-Better-Cheaper programs. He explained that to achieve reliable missions, NASA usually relies on system management techniques in their programs. However, those approaches are not possible for small satellite teams, as they are too time-consuming and expensive. Some failed missions of the Faster-Better-Cheaper program proved that point. For small satellite (and CubeSat) development, the dynamics of the small cohesive teams must be used to achieve reliable satellites, and teamwork rather than formal system management is the key to do that, as also pointed out by McCurdy [14]. This does not mean to disregard every aspect of system management, as for example reviews are highly useful for university-based CubeSat teams. It means that the overhead of large and formal system management cannot be carried by the management team and must be replaced by a less formal set of teamwork methods. In case of reviews, it might be useful to switch from highly structured to less formal reviews, in which problem identification is the main idea rather than producing paperwork, as also suggested by McCurdy [14]. He continued with an example of Mars Pathfinder, in which the core team consisted of about 30 individuals. Small teams can become “self-learning” organizations, thus developing the necessary capacity to perceive risk, solve problems and learn from errors without a formal system management process supporting them. The main requirement for this is to build a strong team, and to invest in team-building throughout the project [14]. As this is true for the teams of the Faster-Better-Cheaper program, it turned also out to be true for the development of MOVE-II. A core team of about 15-20 dedicated students carried the development of the satellite, all of them voluntarily spending more than 1,000 hours in the project. Again we can look into McCurdy’s [14] work how to achieve teamwork such as that: Project leaders have to spend a great amount of time communicating the goals of the mission, or in the case of an university-based project, the benefits of being part of a CubeSat program. This is needed to forge a team in which team members do not perceive

themselves as members of large, multi-layered bureaucracies but as entrepreneurs who come together to solve problems, or in our case to build a CubeSat. In that group it is a question of integrity of everyone to ensure that their hardware and software will work at the end of the day. This also involves the willingness of informally reviewing each other's work and decisions, and most important to expose themselves to hands-on work in hard- and software. This hands-on work will motivate the team members, since they feel personally responsible for the success or failure of the spacecraft [14], [30]. This not only involves hands-on work in designing and producing the spacecraft, it extends to testing the spacecraft and ideally some of the team members are still there for operating it after launch. In MOVE-II, as in teams of Faster-Better-Cheaper, a mixture of experienced and new personnel is also a key to success, as new people can always bring in fresh ideas and view problems from other angles, and experienced students can help new members of the team to grasp the essentials of the developed hard- and software. Furthermore, exposure of team members to multiple areas within the satellite also helps to overcome group thinking and greenwashing within the sub-groups.

Lastly, two underrated characteristics any project manager should embrace in a CubeSat project concern vigilance and communication within the project. An example of vigilance must be set by the project management, since due to the abundance of formal system management processes, teamwork and informal processes must capture and resolve failures when they occur. Also vigilance is needed to assess if the project slowly drifts into the no-fly zone of complexity, schedule and cost [14]. Communication must be another cornerstone of CubeSat teams, as we have shown for our satellite in Section 4.3. McCurdy also presented that small project teams solve reliability problems through informal communication, and thus substitute teamwork for paperwork. This ability is limited by the number of people involved, and the smaller the number, the better team members can resolve problems without relying on a formal process [14]. A lesson learned from MOVE-II is that we might have breached this boundary for some time in the project, as we had times, in which 100 students were involved. As McCurdy pointed out, a 40-person team needs 4-times as many communication channels as a 20-person team [14]. Another important point regarding communication is the ability of the project manager to create trust in his team that any problem can (and should) be put on the table. Galorath & Evans [310] explained in their book that many program managers unconsciously signal that they do not want to hear about any new risk, even if they explicitly communicate it otherwise, thus leading to team members that become reluctant to identify and report risk and failures. We tried to prevent that in MOVE-II by motivating people to find as many flaws as possible in the system, and sometimes put out a small present for the person who found the most bugs within one week. Our experience is substantiated by the successful Mars Pathfinder mission, in which trust was created within the team to bring any problem to the mission management at any time [14].

To conclude the recommendations and the overall discussion, university-built CubeSats have some unique features that must be considered through the development process. In parallel, they share many characteristics with larger missions, especially those working under limited funding and time. For that reason, CubeSat teams can learn from each other but also should have a glance at the lessons learned from larger missions, as no one should repeat failures already made in the past. Table 6-8 in Appendix B summarizes the most important general recommendations from the book of McCurdy [14] for building small satellites and is highly recommended for anyone who intends to manage a university-based CubeSat development project.

6 Conclusion

“Spaceflight will never tolerate carelessness, incapacity, and neglect.”

– Gene Kranz

“The largest obstacle to low-cost innovation is the belief that it cannot be done.”

– Howard E. McCurdy

6.1 Summary

This work was carried out with the intention to increase the chance of MOVE-II to survive its first days and weeks on-orbit. CubeSats are currently suffering from high rates of DOA and infant mortality cases that are often related to poor system level testing before launch. Although CubeSats are already seen as a disruptive innovation for space, this high rate must be reduced before they can fulfill this prediction. The rapid miniaturization of spacecraft, fueled by the consumer market of electronic devices, boosts the development of small satellites. By taking more risk in our missions, and by flying missions more frequently we could reverse the Traditional Space Spiral and get back to a higher pace of innovation in space. Lessons learned of the Faster-Better-Cheaper program of NASA show that low-cost innovation is a rocky path and management decisions largely affect if a low-cost mission will succeed or not. Not all future missions will rely or will be able to rely on CubeSats or small satellites to fulfill their missions. Physical limitations but also the complexity of some mission goals should prevent us from the assumption that everything can be done on a smaller scale in space. Yet the ongoing miniaturization of electronics combined with their high quality will continue to fuel new ideas how to achieve mission goals in the future, and some of them will be carried out by CubeSats.

Spacecraft production has a unique set of properties, since the product itself is often manufactured in a very small, single digit item size or, in some of the cases, is one-of-a-kind, and has to work remotely in a hostile environment. Data on failures can be sometimes communicated through the limited data-channels left on the spacecraft or otherwise have to be subsequently analyzed on-ground with the remaining data. Thus, there are cases in which not every detail on a failure is known and very rarely satellites have been brought back to Earth for later studies. The analysis of historical data showed us that infant mortality was the predominant pattern of failure in the early days of spaceflight. Over the years, software became increasingly important for space missions, and today it is a main source for both on-orbit failures and also failures detected during ground tests. This also held true for MOVE-II. Software failures are by definition design failures and cannot be easily mitigated by traditional approaches such as redundancy. An abundance of engineering failures, such as software errors, will cause spacecraft to fail within the infant mortality zone of the bathtub-curve, and on-orbit data of past missions show exactly that. Despite this, most reliability prediction models assume that spacecraft operate in the constant-failure rate zone of the bathtub curve. If the parts of a system do not work flawlessly together, thus engineering flaws and

workmanship errors are still in the system, this assumption is wrong and can also be not corrected by models that can assume infant mortality and wear-out on part level. Reliability assessments produce test data of subsystems or systems and these data can then be used in reliability growth models, as done in development process of many products for terrestrial markets. The reliability of spacecraft can be assured by multiple, often complementary strategies. Often space-grade parts and redundancies are used in traditional missions. Both is only possible to a very limited degree for CubeSat missions due to their limited resources and envelope (money, mass, volume). The miniaturization and professionalization of the electronics consumer market led to mass-produced electronic components, for which reliability can be assured to a higher confidence than in the limited-scale production of space-grade components. Markets with special environmental needs, such as the automotive market, led to COTS parts that already can cope with most of the environmental challenges of spaceflight. Vacuum and high energy radiation is an exemption from that, but we have shown that through careful testing, part-functionality in vacuum can be assured. For radiation, destructive SEEs must be prevented by design, using suitable circuitry, and the low orbit/mission lifetime of CubeSats missions helps to keep the chance of TID effects on the COTS electronics relatively low. Soft SEEs can be corrected to some degree by most modern consumer electronics with their on-device failure correction algorithms. Currently, radiation is of lesser concern for CubeSats, as very little examples of failures due to radiation are known and many CubeSats fail before the reach a point in life in which they would have a realistic statistical chance to suffer from a catastrophic, radiation-induced error. Nevertheless, this might be a very relevant topic in the near future, and as soon as more CubeSats reach the constant failure rate region of the bathtub-curve, radiation induced errors might be reported more frequently. Currently, due to the high DOA and infant mortality rate, the on-orbit reliability of CubeSats is significantly reduced within the first year. Although CubeSats should be intended to take some risk in their missions, the current rate is too high as almost no useful data are reported back from those early failures. The frequent failures also led to a high rate of CubeSats becoming space debris early in their life. Although it is very valid to discuss the space debris problem in general, and the mitigation of space debris in higher orbits by early failing CubeSats in particular, we have seen that the current space debris problem is caused to a higher degree by larger satellites and only to a lesser degree by CubeSats.

Statistical data of on-orbit failures are useful to deduce patterns of satellite failure but must be carefully used when parametrically fitted. The shape and scale factors of Single-Weibull and 2-Weibull mixture functions determine the fraction of satellites failing due to the different terms over time. Consistency to physical behavior must also be considered when choosing those parameters. Infant mortality is acting at start of life, not throughout the life, otherwise it would not be an effect justifiable by infant mortality. The studied pooled group of satellites of different sizes showed no sign of wear-out as their failure rate function has the shape of a right-open bathtub-curve. The wear-out might be missing due to masking and retirement of satellites before they reach wear-out zone of life. In general, the mixing of different sizes of satellites is considered as not useful and thus the analysis was proceeded with three different groups of satellites. Small satellites (< 500 kg) showed moderate DOA rates and a relatively constant failure after that, while medium satellites (in between 500 and 2500 kg) had a relatively benign infant mortality zone and more pronounced wear-out later in life. Some large satellites (> 2500 kg) also failed directly on arrival or within their early life, but the life of larger satellites is generally dominated by wear-out. We were not able to draw a conclusion which of the three classes of satellites had a higher chance of failure over time, since all three of them arrived at about the same levels of reliability after nine years on-orbit. CubeSat reliability is characterized by very high rates of DOA and infant mortality and a reliability decrease of about 40% within the first year on-orbit can be seen from the studied group of 178 CubeSats. The CubeSat failure database shows that many CubeSats might have failed due to poor system level testing, confirming the work from Swartwout. CubeSats should be able to take more risk the other satellite classes, but the current rate of early failures might forestall their breakthrough as miniaturized, fast tools for space exploration and commercialization. After asking more than 900 individuals affiliated to CubeSats, we found that many developers either

overestimate the chances of success for their own mission, or underestimate the chances others have. This lack of self-perception might also cause riskier approaches than needed.

As of 2018, the Institute of Astronautics and the Technical University of Munich can look back at more than 12 years of hands-on education with CubeSats. Over 70 students gained technical experience through First-MOVE and many important technical and management lessons learned showed the difficulties of building a CubeSat in a university environment. The development of MOVE-II was started in 2015, and as of this writing the satellite should be launched into space in October 2018. Besides the presented technical evolution, we focused on methods to shift our risk upfront and approaches that enable us to conduct subsystem- and system-level testing as early as possible. We built prototypes and so-called brass-boards early and often and thereby mainly followed an approach developed by the AFRL. Agile software development and additive manufacturing helped us to mitigate late risks and we put an easy-to-use failure reporting, analysis, and corrective action system in place to collect and monitor all errors found in system level testing. Whenever possible, we followed a TLYF approach and used the full command chain (mission operations interface – ground station – satellite) while testing. The tracked errors were used to assess our own system, and we also used reliability growth models to project the assessment into the future. Exponential models with and without variable starting date showed the best results in estimating the number of remaining errors in the system. S-Shaped models were of inferior quality, since the projection of remaining errors grew constantly along the number of found errors. A model with a distinction between easy- and difficult-to-find errors was also tested but later discarded since this distinction was not possible for the data collected in MOVE-II. We later filtered the group of collected errors for critical errors that would cause the satellite to fail. We use that group of critical errors for estimations of the launch readiness of MOVE-II, and as of end of 2017, the favored exponential model with variable starting date projected less than a fraction of one failure left in the system. As of this writing, we are continuing the system-level tests of MOVE-II to improve both, the confidence in our satellite and the confidence in our growth models. The filtered group of critical failures was also later used to estimate a reliability of the satellite when put into orbit. Thereby, all projections into the future cannot guarantee that the system will not fail early, since they are dependent on the way the system is tested beforehand. Through heterogeneity of testing vectors and by maximizing the overall system level testing time, we tried to improve our overall chances to find engineering flaws in our system and thus increase our chances of success. Nevertheless, tests that are left out, flawed interactions that are not found, unknown phenomena, and environmental effects not tested on the satellite can always cause early failure. As soon as CubeSats regularly achieve to work in the constant failure rate zone of the bathtub-curve, the presented reliability estimations might be used as an additional parameter for design trade-offs. Since those methods all assume a flawlessly working system, they will all overestimate the reliability of current CubeSats. In general, the presented FIDES approach offers the most promising results, based on-orbit feedback data, and it might be improved in the future if more users are reporting their own on-orbit results. For all reliability prediction methods, we have studied, radiation remains an unknown source of error. We also showed that university-built CubeSats have unique characteristics that have to be considered when managing such projects. The CubeSat community can learn from each other but should also take a glance at lessons learned from larger missions, and especially the Faster-Better-Cheaper program of NASA holds many lessons also applicable for CubeSats. Teamwork must replace formal system management in most cases and strong management is needed to communicate common goals of the mission in order to foster team spirit. Overall, despite their high rate of early failures, the author believes that CubeSat will change the way of how satellites are built in the future, and of how scientific experiments can be carried out. A new generation of enthusiastic space engineers with hands-on experience will emerge from the worldwide university-based CubeSat efforts, and while applying their skills in larger projects, they might be able to reverse the Traditional Space Spiral by utilizing CubeSat and small satellite approaches for parts of their missions. Most important, those people might change the mindset of “can’t fail” for small missions in the future, and carefully evaluate their complexity versus cost and schedule, and this would foremost fuel new ideas for commercial and scientific applications in space.

6.2 Conclusion

The presented results are intended to help to decrease the infant mortality and DOA rates in current CubeSat missions. Focusing on engineering flaws, design errors and software bugs, it is the hope of the author that these kinds of failures will be diminished at some time in the future by using the shown approaches. By focusing on system level testing for a longer period of time, and carefully tracking and subsequently solving the occurring bugs (if possible), the reliability of CubeSats but also the confidence in and the understanding of those systems will hopefully increase. All presented results were achieved on one single satellite, so more data, also from on-orbit, are needed to confirm or disprove our conclusions. The first working hypothesis of this dissertation was:

The time-dependent failure behavior of satellites, namely dead on arrival, infant mortality, random failure and wear-out, can be individually extracted from today's in-flight reliability data.

This working hypothesis was verified using data from Castet & Saleh [120], Saleh & Castet [22] as well as from Dubos et. al [15]. It was shown that the time-dependent failure behavior can be individually extracted, but the parameters of the mixture-Weibull distribution have to be handled with care to not get in conflict with the definition of, e.g. infant mortality. Also, it was shown that the binning of different mass-classes of satellites leads to masking, as certain mass-classes of satellites show different failure behavior as others. The second working hypotheses of this dissertation was:

The failure behavior is substantially different between commercial satellites and CubeSats, and can be quantified.

The second working hypothesis was verified by collecting on-orbit reliability data of CubeSats, building the CFDB and subsequently fitting parametric models to the collected nonparametric data. Different from commercial (larger) satellites, CubeSat reliability is dominated by DOA and infant mortality. The on-orbit reliability of CubeSats is reduced by 40% in their first year, which is a factor of more than 10 to the next bigger category, small satellites, where DOA and infant mortality exist to a far lesser degree. Future missions have to prove if this difference also exists between commercial and non-commercial CubeSats. The last working hypothesis of this dissertation was:

A test strategy for CubeSats can be developed to identify and solve possible DOA and infant mortality causes and thus significantly and efficiently increase their reliability.

The last working hypothesis was also verified. The test strategy, extensively presented in section 4.3, was verified on the CubeSat MOVE-II and helped detecting and solving bugs. The question if this strategy helps to increase the reliability cannot be fully answered yet. Test data on ground shows the maturity of the system and growth models predict the absence of remaining critical bugs. Yet, as long as MOVE-II is not successfully operating in space, it cannot be fully proven that the strategy works. Even then, it remains an open question, since only statistical relevant number of satellites implementing the strategy and showing an increase in reliability to a control group not doing it would be needed to prove it. This might work in theory, but as all satellites implement some kind of testing before launch, it would be a difficult task to decide what to test with the control group and what not. Furthermore, effects stemming from the space environment and spacecraft-unique characteristics might impede any comparison. The one-kind-of characteristic of most current spacecraft allows only limited conclusions on other, different spacecraft, although this might change in the future with the upcoming serial production of satellites.

The following primary objectives for this thesis were defined in Section 3.1:

- 1) Extraction of the time-dependent failure behavior of satellites from today's in-flight reliability data.**
- 2) Collection of CubeSat in-flight reliability data and extraction of the time-dependent failure behavior.**
- 3) Development of a reliability assessment method for CubeSats to identify, track, and subsequently solve possible DOA and infant mortality causes and thus significantly and efficiently increase the reliability of university-built CubeSats.**

The first goal was accomplished by carefully selecting the fitting parameters and subsequently fitting parametric functions to existing non-parametric data of satellites. The complete group of mixed satellite masses showed a right-hand open bathtub-shaped failure rate function, thus no wear-out, in the reliability over time. This could be caused by retirement of jeopardized satellites and/or the binning of different satellite mass classes in one group. Further study of different mass classes revealed a higher susceptibility for infant mortality and DOA in the small satellite class, while medium- and large-sized satellites experienced more pronounced wear-out. At the end of the observation window, all satellite classes showed nearly the same decrease in reliability over time, so no clear relation between mass and reliability was found. While the underlying root causes of the failures were not available for this work, they might help confirm or disprove the patterns we found in the different satellite classes.

The second goal was also achieved, as data from 178 CubeSat missions were collected from publications, various online sources but also from a survey sent out to 978 individuals and these data were subsequently analyzed. CubeSats showed much more pronounced DOA and infant mortality rates than larger satellites, and their reliability drops to 60% after the first year on-orbit. Thereby, the survey was also used to study expectations and mind-sets of CubeSat developers.

The third goal was accomplished, and the developed method was tested on LRT's CubeSat MOVE-II. To tackle the identification, tracking and resolving of bugs, a set of approaches was necessary. A FRACAS system, early prototypes, HiL-testing and accessibility of the satellite via Wi-Fi showed to be the most promising of these approaches to shift risk in the project upfront. Different reliability assessment models were applied on the collected error data and the two most promising models showed the maturity of the system at the end of 2017. 113 critical failures found that were subsequently corrected, significantly increased our chance of success, but also our own confidence in the system. Yet, we continued system level testing in 2018, as segments of the system, especially those on ground, showed no saturation up to the end of 2017. Of course, the presented methods cannot guarantee to 100% that the satellite will work in space after its launch in October 2018, but they can increase the chance of it.

Also, a set of secondary goals was presented in Section 3.1:

- 1) Create a FRACAS that can be used in a university environment to prevent engineering and manufacturing problems from slipping through.**
- 2) Improve the TLYF approach for CubeSats in order to generate accurate data for the system level reliability assessment.**
- 3) Develop a reliability prediction method for CubeSats, to efficiently trade-off design options in early phases.**

The first goal was achieved with the FRACAS, which was developed and later tested in the MOVE-II project. A tracking system established in Redmine helped that no problems slipped through while testing the satellite, and we were able to further improve that by adding an automated bug-tracker called Elfriede. The second goal was achieved by a bundle of approaches, clustered around the easiness-of-test of the satellite and the mindset of “finding as many bugs as possible” rather than “pass/no pass tests”. The already pointed out 24/7 accessibility of both the EM and FM over Wi-Fi was one of the most important approaches for that. Thereby, it was important to hold the motivation of the testers and their numbers high, which was achieved to some degree by competitions on the detection of as many bugs as possible in the system. TLYF also benefited from the early availability of our OPS, which was accomplished by agile software development methods. This early usage of OPS enabled us to operate both models over the complete command chain for a long period of time. Finally, we showed that certain reliability prediction methods might be useful for design trade-offs of CubeSats, but more work has to be done in the future as soon as a majority of CubeSats is not suffering from infant mortality and DOA anymore. Thus, also the third goal was achieved in this work but has to be further expanded in the future.

It is the hope of the author that all presented efforts will also help other university CubeSat teams to increase the chance of success in their missions and also enhance the confidence in their systems. The methods might also help larger satellites, but the applicability in those projects has to be studied in the future.

6.3 Future Work

As already pointed out, the presented work is considered to be a first step towards more reliable CubeSats. Much work was left out and many ideas remain to be explored in the future, either using the presented growth approaches with new satellites or exploring new reliability growth models or entire new ways of assessing the reliability of CubeSats. In the following, future work is presented in sections corresponding to those in Chapter 4.

The analysis of general satellite reliability should be updated in the future, as the used data were from a group of satellites in between 1990 and 2008. 10 years later, there might be other patterns and lessons that can be derived from the collected on-orbit reliability. Also, when building such a new database, further filtering could prove or disprove a correlation between the experience of the satellite manufacturer and on-orbit reliability, as already shown by Swartwout for CubeSats. This filtering could also include the orbit of the spacecraft or a complexity index since both influence the reliability of spacecraft. On-orbit data could also be used as feedback for Bayesian updates of prediction and assessment models, as presented by Ogamba [106]. This is especially interesting for the upcoming higher-volume production of small satellites. Lindsey, Rackley, Brall & Mosleh [311] showed a similar approach, using on-orbit data as prior estimate for a Bayesian reliability prediction model. Data from on-orbit failures can then be parametrically analyzed regarding the three different zones of the bathtub-curve, and different from this work also root-cause data can be used for that purpose. As Conrad [143] showed, having distinct knowledge about the time-dependent on-orbit failure behavior can improve prediction models, as the different failure rate zones from on-orbit feedback can be applied to piece-part failure models. Finally, if on-orbit servicing of satellites is regularly applied at some day in the future, reliability models for repairable systems (see for example Crow [312]) might reveal other strategies needed for the production of spacecraft. These models could already be applied for planned cases of on-orbit servicing and their usefulness thereby studied.

Similar to the update of the general on-orbit reliability data of satellites, an update on the on-orbit data of CubeSats is needed in the future. As before, more filtering could help to find new patterns and relationships, both in the already existing data as well as in the not yet studied group of new CubeSats. Experience of the developer/manufacturer could be one interesting characteristic to research, as already shown by Swartwout in his studies [249]. Also, a further distinction between catastrophic on-orbit errors and minor errors, as for example presented by Saleh & Castet [22] for their studied group of larger satellites, could help reveal

further areas of concern for CubeSats. In general, much work could be done to improve testing of CubeSats and already known weak or yet unknown weak areas of CubeSat reliability. Testing approaches from terrestrial applications, such as highly-automated testing of hard- and software [261], methods from black- and white box testing [313], [314] and methods to improve the state-space coverage [291] could be studied to improve the gain from subsystem- and system-level testing of CubeSats. A general database, similar to the presented European MATED system for larger spacecraft, could help to collect on-orbit failure cases and lessons learned from CubeSat missions. Sharing this information amongst university-based development teams could be possible, and should be valued by everyone, since experience of past failures can influence design decisions and prevent failures from being made over and over again [128]. Further improvement of our FRACAS would help to better collect, distribute and solve problems within CubeSat teams and the filtering could already partly happen in there. Software of CubeSats and software development for CubeSats (but also of larger satellites) will need more attention in the future, as more and more capability is put into software on space missions while using “old-space” software development approaches, such as the V-Model [261]. Agile methods [315] might help to get the satellite’s software ready for first testing at earlier points in time, shifting also this risk upfront. In parallel, project management of CubeSat and small satellite projects might have to change their view and expectations of the software development process in satellite missions, so studying that would be useful as well. The role of software for the mitigation of radiation induced errors [316] in space could also be studied as it is one of the most feasible approaches to achieve at least some radiation tolerance on CubeSats. With upcoming swarms and constellations of CubeSats, studying the collective reliability of those multiple-satellite aggregations would be interesting, both based on the historical on-orbit reliability data but also on projections out of assessments of individual satellites of the group. Engelen, Gill & Verhoeven [317] already showed a first step that could further be expanded in the aforementioned direction. For that, also shared subsystems and collective redundancy could be taken into consideration, both from a technical and from a reliability analysis point of view. On the manufacturing side, this trend could also be very interesting since multi-satellite production enables approaches from terrestrial markets, such as lean production [193]. Thus, the influences and challenges of multi-satellite production would be an interesting field to research. Finally, to implement all of this, also the management methods of projects such as ours might be feasible to study and to advance in the future. The lessons learned of Faster-Better-Cheaper showed us the important role of project management for the development of small spacecraft. One lesson already learned for CubeSats that might expand to the small satellite domain in the nearby future is on the value of standardization of satellites. Work on further expanding the CubeSat standard to larger sizes around 50 kg or 100 kg could help to fuel the rapid innovation in space.

All presented work on reliability assessment and prediction must be evaluated against on-orbit data not only of MOVE-II, but preferably from several other missions to get statistically significant test and on-orbit feedback. Also, as already pointed out, the FRACAS could be improved in the future to allow better filtering of error data. Examples for that are, to collect data on the difficulty of error detection for certain growth models, to try to determine the Fix Effectiveness Factor needed for enhanced AMSAA-Crow models [176], or more detailed information on the root-causes of and reasons for the errors. Besides on-orbit feedback, reliability growth modelling of CubeSats could be enhanced by the experience of other CubeSats using the proposed assessment while system-level testing their satellites. On the modelling side, the aforementioned Crow-AMSAA models [176] and inverse Weibull functions [181] could help to better predict the remaining errors in the system and the reliability growth over time. While viewing CubeSats as software-dominated systems, logistic and Gompertz growth curves [180] or other software reliability growth models [318] could be also used for improvement of the estimation. Bayesian models could also help for that purpose and they furthermore allow the incorporation of prior knowledge, thus data that stem from other sources than the system level tests of the CubeSat. An example for this could be on-orbit data from existing similar satellites. Future prediction models could also be improved by both, on-orbit feedback and failure rates from test data, using some kind of correction factor as already proposed by Hecht [54]. Ideally, future CubeSats

could also work with a so-called reliability budget, as presented by Goel & Graves [40] and Hecht [13]. Although initially seen as a method to account for wear-out in the analysis, this could help to directly integrate results of system level testing into reliability predictions. Ideally, this would also influence cost models of the spacecraft, as proposed by Hecht [13].

The advent of Mega-constellations and large networks of miniaturized satellites will possibly change many approaches currently used in spacecraft manufacturing and reliability modelling/engineering of satellites. Historical examples, such as the assembly, integration and testing of the spacecraft of the Globalstar constellation [319] showed that multi-spacecraft manufacturing not only changes test campaigns on ground, it also influences manufacturing and testing due to possible feedback from on-orbit results of already launched satellites. Spacecraft swarms and constellations might enable us to use functional redundancy across different satellites, and this approach should be investigated in the future. Although higher production rates of satellites historically showed that early failures decrease over time mainly due to extinction of design errors, as shown for the TRDS and GOES satellites [183], research has to be carried out what kind of system level tests can be left out for “mass-produced” satellites. Manufacturing errors and other engineering flaws could still persist in single satellites, and the risk of creating a runaway effect in a very populated orbit is very high for the planned Mega-constellations. CubeSats could help to easily test and verify hard- and software of those planned constellations of larger satellites and thus make their contribution to reverse the Traditional Space Spiral, but also change the mindset of “can’t fail” for small space missions. To do that, the current DOA and infant mortality rates must be improved. Developing and testing a CubeSat, but also accepting the inherent risk of failure of those small satellites, demands passion from those who are involved. Setbacks and the inherent risk CubeSats have to carry must be boldly accepted, and while building, launching and operating those small, innovative missions we always have to remind ourselves that:

“Through endurance we conquer.”

– Ernest Shackleton

7 References

- [1] B. Twiggs, "Origin of CubeSat," in *Small satellites: Past, present, and future*, H. Helvajian, Ed., El Segundo Calif.: Aerospace Press, 2008, pp. 151–173.
- [2] National Academies of Sciences, Engineering, and Medicine, "Achieving Science with CubeSats: Thinking Inside the Box", Washington, DC: The National Academies Press, 2016.
<https://doi.org/10.17226/23503>.
- [3] G. Martin, "NewSpace: The Emerging Commercial Space Industry," ISU Master Class Lecture; Feb. 2017; Strasbourg; France.
- [4] H. Stoewer, "Future of Space: Evolution or Revolution?," ArianeGroup 8th R&T Days, Nov. 2017, Paris, France.
- [5] M. Swartwout, *CubeSat Database*. [Online] Available:
<https://sites.google.com/a/slu.edu/swartwout/home/cubesat-database>. Accessed on: Feb. 27 2018.
- [6] D. Werner, *NOAA sees great promise and challenges in using data from small satellite constellations - SpaceNews.com*. [Online] Available: <http://spacenews.com/noaa-smallsat/>. Accessed on: Jan. 09 2018.
- [7] Centaur Communications Ltd, *Contract heralds "new paradigm in weather forecasting"* [Online] Available: <https://www.theengineer.co.uk/contract-weather-satellite/>. Accessed on: Feb. 16 2018.
- [8] C. R. Boshuizen, J. Mason, P. Klupar, and S. Spanhake, "Results from the Planet Labs Flock Constellation," Proceedings of the 2014 AIAA/USU Conference on Small Satellites, Private Endeavors, SSC14-I-1. <http://digitalcommons.usu.edu/smallsat/2014/PrivEnd/1/>.
- [9] S. Spangelo, J. Castillo-Rogez, A. Frick, A. Klesh, and B. Sherwood, "JPL's Advanced CubeSat Concepts for Deep Space Exploration," CubeSat Workshop, Logan, Utah, Aug. 2015.
- [10] M. Sweeting, "Small Satellites: quo vadis?," AIAA SPACE 2012 Conference & Exposition, Sep. 2012, Pasadena, California, USA.
- [11] M. Swartwout, "The First One Hundred CubeSats: A Statistical Look," *Journal of Small Satellites*, 2013, Vol. 2, No. 2, pp. 213-233.
- [12] M. Langer and J. Bouwmeester, "Reliability of CubeSats – Statistical Data, Developers' Beliefs and the Way Forward," Proceedings of the 30th Annual AIAA/USU Conference on Small Satellites, Logan, UT, 6-11 August, 2016, Paper SSC16-X-2.
- [13] H. Hecht, "Reliability During Space Mission Concept Exploration," in *Space mission analysis and design*, W. J. Larson and J. R. Wertz, Eds., 2nd ed.: Kluwer Academic Publishers, 1998, pp. 700–714.
- [14] H. E. McCurdy, *Faster, better, cheaper: Low-cost innovation in the U.S. space program*. Baltimore, Md., London: Johns Hopkins University Press, 2003.
- [15] G. F. Dubos, J.-F. Castet, and J. H. Saleh, "Statistical Reliability Analysis of Satellites by Mass Category: Does Spacecraft Size Matter?," in *60th International Astronautical Congress 2009 (IAC 2009): Daejeon, Republic of Korea, 12 - 16 October 2009*, IAC-09-D1.3.6.
- [16] A. Pasztor, *How to Build Satellites Much Faster—and Cheaper*. The Wall Street Journal, June 2016. Accessed on: Feb. 27 2018.
- [17] R. A. Bauer, P. S. Millar, and C. D. Norton, "Bridging the Technology Readiness "Valley of Death" Utilizing Nanosats," 21st Ka and Broadband Communications Conference; Oct. 2015; Bologna; Italy.
- [18] M. Czech, "In-Orbit versus Ground Testing – Analysis Framework for Evaluation," Dissertation, Technical University of Munich, 2016.

- [19] J. N. Pelton, "Lifetime Testing, Redundancy, Reliability, and Mean Time to Failure," in *Handbook of Satellite Applications*, J. N. Pelton, S. Madry, and S. Camacho-Lara, Eds., New York, NY: Springer New York, 2016, pp. 1–18.
- [20] J. Abel, "Quick-Turn, Low Cost Spacecraft Development Principles," Proceedings of the 30th Annual AIAA/USU Conference on Small Satellites, Logan, UT, 6-11 August, 2016.
- [21] M. J. Hecht and H. Hecht, "Reliability," in *Space mission engineering: The new SMAD*, J. R. Wertz, D. F. Everett, and J. J. Puschell, Eds., Hawthorne, CA: Microcosm Press, 2011, pp. 753–767.
- [22] J. H. Saleh and J.-F. Castet, *Spacecraft Reliability and Multi-State Failures*. Chichester, UK: John Wiley & Sons, Ltd, 2011.
- [23] S. L. Hogan, *Effective Fault Management Guidelines: AEROSPACE REPORT NO. TOR-2009 (8591)-14, The Aerospace Corporation*, 2009.
- [24] Department of Defense, "Military Handbook: Reliability Prediction of Electronic Equipment," MIL-HDBK-217F, Dec. 1991.
- [25] Department of Defense, "Military Handbook: Reliability Prediction of Electronic Equipment," MIL-HDBK-217F, NOTICE 2, Feb. 1995.
- [26] S. A. McDermott, A. Jacobovits, and H. Yashiro, "Automotive electronics in space: Combining the advantages of high reliability components with high production volume," in *2002 IEEE Aerospace Conference proceedings: Big Sky, Montana, March 9-16, 2002*, Big Sky, MT, USA, 2002, 4-1857-4-1869.
- [27] R. Fleeter, "Design of Low-Cost Spacecraft," in *Space technology library*, vol. 8, *Space mission analysis and design*, W. J. Larson and J. R. Wertz, Eds., 3rd ed., Torrance, Calif.: Microcosm Press, 1999, pp. 853–882.
- [28] L. Sarsfield, *The cosmos on a shoestring: Small spacecraft for space and earth science*. Santa Monica CA: RAND Critical Technologies Institute, 1998.
- [29] J. F. Binkley, P. G. Cheng, P. L. Smith, and W. F. Tosney, "From Data Collection to Lessons Learned—Space Failure Information Exploitation at te Aerospace Corporation," in *Proceedings of the First International Forum on Integrated System Health Engineering and Management*, Napa, CA, 2005.
- [30] J. R. Wertz, "Reducing Space Mission Cost and Schedule," in *Space mission engineering: The new SMAD*, J. R. Wertz, D. F. Everett, and J. J. Puschell, Eds., Hawthorne, CA: Microcosm Press, 2011, pp. 355–366.
- [31] J. C. New and A. R. Timmins, "Effectiveness of Environment-Simulation Testing for Spacecraft," NASA Technical Note TN D-4009, Jun. 1967.
- [32] J. H. Boeckel, A. R. Timmins, and K. R. Mercy, "Goddard Space Flight Center Test Philosophy and Resultant Record," NASA Technical Note TN D-5812, Jul. 1970.
- [33] D.N.P. Murthy, M. Rausand, and S. Virtanen, "Investment in new product reliability," *Reliability Engineering & System Safety*, vol. 94, no. 10, pp. 1593–1600, 2009.
- [34] P. B. de Selding, *Airbus and OneWeb form joint venture to build 900 satellites*. Space News, Jan 2016. [Online] Available: <http://spacenews.com/airbus-and-oneweb-form-joint-venture-to-build-900-satellites/>. Accessed on: Mar. 09 2018.
- [35] J. Gonzalo, D. Domínguez, and D. López, "On the challenge of a century lifespan satellite," *Progress in Aerospace Sciences*, vol. 70, pp. 28–41, 2014.
- [36] R. H. Maurer, "Spacecraft reliability, quality assurance, and radiation effects," in *Johns Hopkins University Applied Physics Laboratory Series in Science and Engineering, Fundamentals of Space Systems*, V. L. Pisacane, Ed., 2nd ed., New York: Oxford University Press, Incorporated, 2005.
- [37] W. F. Tosney and S. Pavlica, "Satellite Verification Planning: Best Practices and Pitfalls Related to Testing," Proceedings of the 5th International Symposium on Environmental Testing for Space Programmes Jun. 2004, Noordwijk, The Netherlands (ESA SP-558, Aug. 2004).

- [38] M. S. Hurley and W. E. Purdy, "Designing and Managing for a Reliability of Zero," Proceedings of the Small Satellites Systems and Services – The 4S Symposium 2010, Jun. 2010, Funchal, Madeira, Portugal.
- [39] A. Birolini, *Reliability Engineering: Theory and Practice*, 7th ed. Berlin, Heidelberg, s.l.: Springer Berlin Heidelberg, 2014.
- [40] A. Goel and R. J. Graves, "Electronic System Reliability: Collating Prediction Models," *IEEE Trans. Device Mater. Reliab.*, vol. 6, no. 2, pp. 258–265, 2006.
- [41] Powell, Harry, R., "The Impact of Reliability on Design," paper No. 60-MD-2, American Soc. of Mechanical Engineers, 1960.
- [42] T. C. Reeves, *Reliability Prediction - Its Validity and Application as a Design Tool*: American Soc. of Mechanical Engineers, paper No. 60-MD-1, 1960.
- [43] S. Clark, *Planet Labs takes rash of launch failures in stride – Spaceflight Now*. [Online] Available: <https://spaceflightnow.com/2015/09/20/planet-labs-takes-rash-of-launch-failures-in-stride/>. Accessed on: Jan. 09 2018.
- [44] European Power Supply Manufacturers Association (EPSMA), "Reliability: Guidelines to Understanding Reliability Prediction," Jun. 2005.
- [45] H. Rinne, *The Weibull distribution: A handbook*. Boca Raton, Fla.: Chapman & Hall/CRC, 2009.
- [46] H. Wilker, *Weibull-Statistik in der Praxis: Leitfaden zur Zuverlässigkeitsermittlung technischer Produkte*. Norderstedt: Books on Demand, 2004.
- [47] R. B. Abernethy, *The new Weibull handbook: Reliability & statistical analysis for predicting life, safety, survivability, risk, cost and warranty claims*, 4th ed. North Palm Beach, Fla.: Abernethy, 2005.
- [48] W. R. Blischke and D. N. P. Murthy, *Reliability: Modeling, prediction, and optimization*. New York, NY: Wiley, 2000.
- [49] ReliaSoft Publishing, *Weibull Distribution: Characteristics of the Weibull Distribution*. [Online] Available: <http://www.weibull.com/hotwire/issue14/relbasics14.htm>. Accessed on: Mar. 12 2018.
- [50] K. L. Bedingfield, R. D. Leach, and M. B. Alexander, "Spacecraft System Failures and Anomalies Attributed to the Natural Space Environment," NASA Reference Publication 1390, Aug. 1996.
- [51] D. A. Galvan, B. Hemenway, D. Baiocchi, and W. Welser, *Satellite anomalies: Benefits of a centralized anomaly database and methods for securely sharing information among satellite operators*. Santa Monica, CA: RAND, 2014.
- [52] R. Gaillard, "Single Event Effects: Mechanisms and Classification," in *Frontiers in Electronic Testing, Soft Errors in Modern Electronic Systems*, M. Nicolaidis, Ed., Boston, MA: Springer US, 2011, pp. 27–54.
- [53] F. Sturesson, "Single Event Effects (SEE) Mechanism and Effects," Space Radiation and its Effects on EEE Components, EPFL Space Center Jun. 2009.
- [54] H. Hecht and M. J. Hecht, "Reliability Prediction for Spacecraft," RADC Report RADC-TR-85-229. Rome Air Development Center, NY: Department of Defense, Dec. 1985.
- [55] P. G. Cheng, "How Software Errors Contribute to Satellite Failures: Challenges Facing the Risk Analysis Community," SCSRA Annual Workshop, May. 2003.
- [56] E. E. Euler and S. D. Jolly, "The Failures of the Mars Climate Orbiter and Mars Polar Lander: A Perspective from the People Involved," Proceedings of 24th Annual AAS Guidance and Control Conference, Feb. 2001, Breckenridge, Colorado, USA.
- [57] J. L. Lions, "Ariane 5 Flight 501 Failure: Report of the Inquiry Board," Paris, France, Jul. 1996.
- [58] JPL Special Review Board, "Report on the Loss of the Mars Polar Lander and Deep Space 2 Missions," JPL D-18709, Mar. 2000.
- [59] M. R. Lowry, "Software Construction and Analysis Tools for Future Space Missions," in *Lecture Notes in Computer Science, Tools and Algorithms for the Construction and Analysis of Systems*, G. Goos, J. Hartmanis, J. van Leeuwen, J.-P. Katoen, and P. Stevens, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 1–19.

- [60] N. G. Leveson, "Role of Software in Spacecraft Accidents," *Journal of Spacecraft and Rockets*, vol. 41, no. 4, pp. 564–575, 2004.
- [61] N. G. Leveson, "Software Challenges In Achieving Space Safety," *Journal of the British Interplanetary Society*, no. 62, July, Aug. 2009.
- [62] P. A. Judas and L. E. Prokop, "A historical compilation of software metrics with applicability to NASA's Orion spacecraft flight software sizing," *Innovations Syst Softw Eng*, vol. 7, no. 3, pp. 161–170, 2011.
- [63] D. Dvorak, "NASA Study on Flight Software Complexity," in *Infotech@Aerospace Conferences, AIAA Infotech@Aerospace Conference*, Seattle, Washington, USA, 2009.
- [64] B. Arnheim, "Lessons Learned from Space Vehicle Test Trends," in *Proceedings of 23rd Aerospace Testing Seminar 2006*, Manhattan Beach, California, USA, Oct. 2006.
- [65] J. Alonso, M. Grottke, A. P. Nikora, and K. S. Trivedi, "The Nature of the Times to Flight Software Failure during Space Missions," in *IEEE 23rd International Symposium on Software Reliability Engineering (ISSRE): 27 - 30 Nov. 2012*, Dallas, TX, USA.
- [66] M. Grottke, R. Matias, and K. S. Trivedi, "The fundamentals of software aging," in *IEEE International Conference on Software Reliability Engineering workshops*, Seattle, WA, USA, 2008.
- [67] M. Grottke, A. P. Nikora, and K. S. Trivedi, "An empirical investigation of fault types in space mission system software," in *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2010*, Chicago, IL, USA, 2010.
- [68] J. Alonso, M. Grottke, A. P. Nikora, and K. S. Trivedi, "An empirical investigation of fault repairs and mitigations in space mission system software," in *43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Budapest, Hungary, 2013.
- [69] J. Nieberding, "Space System Development: Lessons Learned," Conference on Quality in the Space and Defense Industries, Mar. 2011, Cape Canaveral, Florida, USA.
- [70] L. Bianchi, "New Reliability Prediction Methodology Aimed at Space Applications: Briefing Meeting with Industry, ESA/ESTEC Product Assurance and Safety Department," Apr. 2016.
- [71] L. S. Swenson Jr, J. M. Grimwood, and C. C. Alexander, "This New Ocean: A History of Project Mercury. NASA SP-4201," *NASA Special Publication*, vol. 4201, 1966.
- [72] C. F. Willard, "Satellite Reliability Spectrum, Interim Report," ARINC Research Corporation, Contract SD-77, Publication No. 173-3-255, 1961.
- [73] C. E. Bloomquist *et al.*, "Operational Reliability Assessment of the GEOS A Spacecraft," NASA Technical Advisement Memorandum No 106-10, 1965.
- [74] Planning Research Corporation, "Reliability Assessment of the GEOS A spacecraft," Technical Report PRC R-760, 1965.
- [75] A. R. Timmins, "The effectiveness of systems tests in attaining reliable earth satellite performance," NASA Technical Note TN D-3713, 1966.
- [76] K. R. Mercy, "Environmental Test Contribution to Spacecraft Reliability," NASA Technical Note TN D-4181, 1967.
- [77] F. R. Wright, "Failure Rate Computations Based on Mariner Mars 1964 Spacecraft Data," NASA Technical Report 32-1036, 1967.
- [78] J. H. Boeckel and A. R. Timmins, "Test Plan Optimization for an Explorer-Size Spacecraft," NASA Technical Note TN D-5283, 1969.
- [79] A. R. Timmins and R. E. Heuser, "A Study of First-Day Space Malfunctions," NASA Technical Note TN D-6474, 1971.
- [80] A. R. Timmins, "A Study of First-Month Space Malfunctions," NASA Technical Note TN D-7750, 1974.
- [81] A. R. Timmins, "A Study of the Total Space Life Performance of Goddard Spacecraft," NASA Technical Note TN D-8017, 1975.

- [82] A. R. Timmins, R. E. Heuser, and J. C. Strain, "Analysis of Flight Model Spacecraft Performance During Thermal-Vacuum Tests," NASA Technical Note TN D-7408, 1973.
- [83] H. P. Norris and A. R. Timmins, "Failure Rate Analysis of Goddard Space Flight Center Spacecraft Performance During Orbital Life," NASA Technical Note TN D-8272, Jul. 1976.
- [84] C. E. Bloomquist and W. C. Graham, "Analysis of Spacecraft Anomalies," NASA-CR-137854, 1976.
- [85] S. E. Levine, "A Review of the Mission Success of Communications Satellites and Related Spacecraft," Aerospace Report TR-0078(3417)-1, 1977.
- [86] J. C. Baker and G. A. Baker Sr., "Impact of the space environment on spacecraft lifetimes," *Journal of Spacecraft and Rockets*, vol. 17, no. 5, pp. 479–480, 1980.
- [87] E. F. Shockey, "A Study of the Longevity and Operational Reliability of Goddard Spacecraft: 1960-1980," NASA Technical Memorandum 82178, Aug. 1981.
- [88] B. W. Augenstein, "Rand Spacecraft Acquisition Study: A Briefing," WN-9551-PR, Aug. 1976.
- [89] F. Barnett, "Demonstrated Orbital Reliability of TRW Spacecraft," TRW publication 74-2286.142, Dec. 1974.
- [90] C. E. Bloomquist and W. C. Graham, "Analysis of Spacecraft On-Orbit Anomalies and Lifetimes," NASA-CR-170565, 1983.
- [91] C. E. Bloomquist, "Spacecraft Anomalies and Lifetimes," Proceedings of the Annual Reliability and Maintainability Symposium, 1984.
- [92] T. L. Ferguson and T. H. Davey, "Analysis of Orbital Satellite Storage," Space Divison, Air Force Systems Command, Report SD-TR-86-02, 1985.
- [93] M. Hecht and E. Fiorentino, "Causes and effects of spacecraft failures," *Qual. Reliab. Engng. Int.*, vol. 4, no. 1, pp. 11–20, 1988.
- [94] C. E. Ebeling, "Parametric Estimation of R&M Parameters During the Conceptual Design of Space Vehicles," pp. 955–959, Proceedings of the IEEE 1992 National Aerospace and Electronics Conference, 1992.
- [95] J. L. Stevenson and R. Strauss, "The operational reliability of the Intelsat V satellite fleet," 14th International Communication Satellite Systems Conference and Exhibit, Washington, DC, USA, 1992.
- [96] R. Sperber, "Better with age and experience - Observed satellite in-orbit anomaly rates," in *International Communications Satellite Systems Conferences (ICSSC), 15th International Communications Satellite Systems Conference and Exhibit: American Institute of Aeronautics and Astronautics*, 1994.
- [97] M. Krasich, "Reliability Prediction Using Flight Experience - Weibull Adjusted Probability of Survival, WAPS," NASA Technical Report, Document ID: 20060041898, Jet Propulsion Laboratory, April 1995.
- [98] G. E. Hall and P. T. Blay, "Early Orbit Space Vehicle Failures - Lessons," *IFAC Proceedings Volumes*, vol. 29, no. 1, pp. 7492–7497, 1996.
- [99] B. R. Sullivan and D. L. Akin, "A Survey of Serviceable Spacecraft Failures," AIAA Space 2001 Conference and Exposition. Albuquerque, NM, U.S.A.
- [100] B. Robertson and E. Stoneking, "Satellite GN&C Anomaly Trends," in *Proceedings of the 26th annual AAS Guidance and Control Conference: AAS 03-071*, Breckenridge, Colorado, Feb. 2003.
- [101] A. R. Hoffman, N. W. Green, and H. B. Garrett, "Assessment of In-Flight Anomalies of Long Life Outer Planet Missions," Proceedings of the 5th International Symposium on Environmental Testing for Space Programmes, Jun. 2004, Noordwijk, The Netherlands (ESA SP-558, Aug. 2004).
- [102] D. M. Harland and R. D. Lorenz, *Space Systems Failures: Disasters and Rescues of Satellites, Rockets and Space Probes*. New York: Springer Praxis Publishing Ltd, 2005.
- [103] N. Green, A. Hoffman, T. Schow, and H. Garrett, "Anomaly Trends for Robotic Missions to Mars: Implications for Mission Reliability," in *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 2006.

- [104] M. Tafazoli, "A study of on-orbit spacecraft failures," *Acta Astronautica*, vol. 64, no. 2-3, pp. 195–205, 2009.
- [105] J. A. Rodiek and H. W. Brandhorst Jr., *Solar Array Reliability in Satellite Operations*, Proceedings of 33rd IEEE Photovoltaic Specialists Conference, San Diego, California, USA 2008.
- [106] N. U. Ogamba, "A Parametric Reliability Prediction Tool for space applications," in *Annual Reliability and Maintainability Symposium (RAMS)*, Fort Worth, TX, USA, 2009, pp. 195–200.
- [107] M. S. Hurley Jr. and W. E. Purdy, "Cost vs. Reliability - Focusing on the Mission Objectives," in *Space mission engineering: The new SMAD*, J. R. Wertz, D. F. Everett, and J. J. Puschell, Eds., Hawthorne, CA: Microcosm Press, 2011, pp. 366–375.
- [108] L. N. Monas, J. Guo, and E. K. A. Gill, "Small Satellite Reliability Modeling: A Statistical Analysis," Proceedings of the Small Satellites Systems and Services – The 4S Symposium 2012, Jun. 2012, Portoroz, Slovenia.
- [109] J. Guo, L. Monas, and E. Gill, "Statistical analysis and modelling of small satellite reliability," *Acta Astronautica*, vol. 98, pp. 97–110, 2014.
- [110] J. Guo, J. Kolmas, and E. K. A. Gill, "Small Satellite Reliability Research on Spacecraft under 50 kg: Analysis on Component Level," Proceedings of the Small Satellites Systems and Services – The 4S Symposium 2014, May. 2014, Porto Petro, Majorca, Spain.
- [111] A. Gorbenko, V. Kharchenko, O. Tarasyuk, and S. Zasukha, "A Study of Orbital Carrier Rocket and Spacecraft Failures: 2000-2009," pp. 179–198, *Information & Security*, Vol. 28, No. 2, 2012,
- [112] G. Fox, R. Salazar, H. Habib-Agahi, and G. F. Dubos, "A Satellite Mortality Study to Support Space Systems Lifetime Prediction," Proceedings of the IEEE Aerospace Conference, Big Sky, Montana, USA, 2013.
- [113] C. Palla, M. Peroni, and J. Kingston, "Failure analysis of satellite subsystems to define suitable de-orbit devices," *Acta Astronautica*, vol. 128, pp. 343–349, 2016.
- [114] M. Peroni *et al.*, "Reliability study for LEO satellites to assist the selection of end of life disposal methods," in *Proceedings of 3rd IEEE International Workshop on Metrology for Aerospace: Florence, Italy, June 21-23, 2016*, pp. 141–145.
- [115] J.-F. Castet and J. H. Saleh, "Satellite Reliability: Statistical Data Analysis and Modeling," AIAA 2009-6556, AIAA SPACE 2009 Conference & Exposition, Pasadena, California, Sep. 2009.
- [116] J.-F. Castet and J. H. Saleh, "Satellite Reliability: Statistical Data Analysis and Modeling," *Journal of Spacecraft and Rockets*, vol. 46, no. 5, pp. 1065–1076, 2009.
- [117] J.-F. Castet and J. H. Saleh, "Satellite and satellite subsystems reliability: Statistical data analysis and modeling," *Reliability Engineering & System Safety*, vol. 94, no. 11, pp. 1718–1728, 2009.
- [118] G. F. Dubos, J.-F. Castet, and J. H. Saleh, "Statistical reliability analysis of satellites by mass category: Does spacecraft size matter?," *Acta Astronautica*, vol. 67, no. 5-6, pp. 584–595, 2010.
- [119] T. Hiriart, J.-F. Castet, J. M. Lafleur, and J. H. Saleh, "Comparative Reliability of GEO, LEO, and MEO Satellites," in *60th International Astronautical Congress 2009 (IAC 2009): Daejeon, Republic of Korea, 12 - 16 October 2009*, IAC-09.D1.6.1.
- [120] J.-F. Castet and J. H. Saleh, "Single versus mixture Weibull distributions for nonparametric satellite reliability," *Reliability Engineering & System Safety*, vol. 95, no. 3, pp. 295–300, 2010.
- [121] J.-F. Castet and J. H. Saleh, "Spacecraft Technologies: Satellite Reliability," in *Encyclopedia of aerospace engineering*, R. Blockley and W. Shyy, Eds., Chichester West Sussex U.K., Hoboken N.J.: Wiley, 2010, 4435-4446.
- [122] J.-F. Castet and J. H. Saleh, "Beyond reliability, multi-state failure analysis of satellite subsystems: A statistical approach," *Reliability Engineering & System Safety*, vol. 95, no. 4, pp. 311–322, 2010.
- [123] J.-F. Castet and J. H. Saleh, "On the concept of survivability, with application to spacecraft and space-based networks," *Reliability Engineering & System Safety*, vol. 99, pp. 123–138, 2012.

- [124] W. Peng and H. Zhang, "Satellite Reliability Modeling With Modified Weibull Extension Distribution," International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (ICQR2MSE), Jun. 2012, Chengdu, China.
- [125] M. Xie, Y. Tang, and T. N. Goh, "A modified Weibull extension with bathtub-shaped failure rate function," *Reliability Engineering & System Safety*, vol. 76, no. 3, pp. 279–285, 2002.
- [126] J. K. Wayer, J.-F. Castet, and J. H. Saleh, "Spacecraft attitude control subsystem: Reliability, multi-state analyses, and comparative failure behavior in LEO and GEO," *Acta Astronautica*, vol. 85, pp. 83–92, 2013.
- [127] J. H. Saleh, F. Geng, M. Ku, and M. L.R. Walker II, "Electric propulsion reliability: Statistical analysis of on-orbit anomalies and comparative analysis of electric versus chemical propulsion failure rates," *Acta Astronautica*, vol. 139, pp. 141–156, 2017.
- [128] P. Cheng and P. Smith, "Learning from Other People's Mistakes," *Crosslink Fall 2007*, pp. 20–24.
- [129] M. Rufer, "Intangible Factors in Manufacturing," in *Space mission engineering: The new SMAD*, J. R. Wertz, D. F. Everett, and J. J. Puschell, Eds., Hawthorne, CA: Microcosm Press, 2011, pp. 743–750.
- [130] J. H. Saleh and K. Marais, "Reliability: How much is it worth? Beyond its estimation or prediction, the (net) present value of reliability," (en), *Reliability Engineering & System Safety*, vol. 91, no. 6, pp. 665–673, 2006.
- [131] A. Goel and R. Graves, "Assessing the Assumptions Used in Reliability Prediction Modeling," in *1st Electronics Systemintegration Technology Conference, 2006*, Dresden, Germany, Sep. 2006, pp. 1143–1148.
- [132] C. Johansson, "On System Safety and Reliability Methods in Early Design Phases: Cost Focused Optimization Applied on Aircraft Systems," Linköping Studies in Science and Technology, Thesis No. 1600, 2013.
- [133] J. Jones and J. Hayes, "Estimation of system reliability using a "non-constant failure rate" model," *IEEE Trans. Rel.*, vol. 50, no. 3, pp. 286–288, 2001.
- [134] P. D. T. O'Connor, "Quantifying uncertainty in reliability and safety studies," *Microelectronics Reliability*, vol. 35, no. 9, pp. 1347–1356, 1995.
- [135] J. McLeish and W. Tomczykowski, "An Introduction to Physics Failure and Reliability Physics Methods," Proceedings of the 2013 Annual RELIABILITY and MAINTAINABILITY Symposium, Jan. 2013, Orlando, FL, USA.
- [136] W. Gericke *et al.*, "A methodology to assess and select a suitable reliability prediction method for EEE components in space applications," in *Proceedings of the European Space Components Conference, ESCCON, 2002*, Toulouse, France.
- [137] G. P. Pandian, D. Das, C. Li, E. Zio, and M. Pecht, "A critique of reliability prediction techniques for avionics applications," *Chinese Journal of Aeronautics*, vol. 31, no. 1, pp. 10–20, 2018.
- [138] National Research Council (U.S.); National Academies Press (U.S.), *Reliability growth: Enhancing defense system reliability*. Washington D.C.: The National Academies Press, 2015.
- [139] P. Carton, M. Giraudeau, and F. Davenel, *New FIDES Models for Emerging Technologies*, Proceedings of the Annual Reliability and Maintainability Symposium (RAMS) 2017.
- [140] P. D. T. O'Connor and A. Kleyner, *Practical Reliability Engineering*. Chichester, UK: John Wiley & Sons, Ltd, 2011.
- [141] J.-F. Castet and J. H. Saleh, "Stochastic Petri Nets for System Survivability and Multi-State Failure Analyses with Application to Space Systems," in *52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Denver, Colorado, 2011.
- [142] A. C. Owens and O. L. de Weck, "Use of Semi-Markov Models for Quantitative ECLSS Reliability Analysis: Spares and Buffer Sizing," ICES-2014-116, Proceedings of the 44th International Conference on Environmental Systems Jul. 2014, Tucson, Arizona.

- [143] D. A. Conrad, "Estimation of satellite lifetime from orbital failure experience," *Journal of Spacecraft and Rockets*, vol. 13, no. 2, pp. 75–81, 1976.
- [144] J. B. Binckes, "Satellite Reliability Estimation: Past and Present Procedures," in *NATO ASI Series, Series F*, vol. 3, *Electronic Systems Effectiveness and Life Cycle Costing*, J. K. Skwirzynski, Ed., Berlin, Heidelberg: Springer, 1983, pp. 333–355.
- [145] G. Krebs, *Intelsat-4*. [Online] Available: http://space.skyrocket.de/doc_sdat/intelsat-4.htm. Accessed on: Mar. 22 2018.
- [146] J. J. Marin and R. W. Pollard, *Experience Report on the FIDES Reliability Prediction Method*, Proceedings of the Annual Reliability and Maintainability Symposium 2005, Alexandria, Virginia, USA.
- [147] M. Zahran, S. Tawfik, and G. Dyakov, "L.E.O. Satellite Power Subsystem Reliability Analysis," *Journal of Power Electronics*, Vol. 6, No. 2, Apr. 2006.
- [148] J. C. Burke and J. W. Evans, "Reliability of wear-out items in electric motors in space — A case study," in *Proceedings / Annual Reliability and Maintainability Symposium (RAMS)*, Jan. 2011, Lake Buena Vista, FL, USA, pp. 1–7.
- [149] J. Wu, S. Yan, and L. Xie, "Reliability analysis method of a solar array by using fault tree analysis and fuzzy reasoning Petri net," *Acta Astronautica*, vol. 69, no. 11-12, pp. 960–968, 2011.
- [150] J. Wu, S. Yan, L. Xie, and P. Gao, "Reliability apportionment approach for spacecraft solar array using fuzzy reasoning Petri net and fuzzy comprehensive evaluation," *Acta Astronautica*, vol. 76, pp. 136–144, 2012.
- [151] R. Witt, A. Kennedy, B. Baetz, U. Mohr, and J. Eickhoff, "Implementation of Fault Management Capabilities for the Flying Laptop Small Satellite Project through a Failure-Aware System Model," AIAA 2013-4661, Guidance, Navigation, and Control and Co-located Conferences Aug. 2013, Boston, MA.
- [152] M. Kaminsky, L. Gallo, and J. W. Evans, "A Bayesian Framework for Reliability Analysis of Spacecraft Deployments," Proceedings of the 2013 IEEE Aerospace Conference; Mar. 2013; Big Sky, MT; United States.
- [153] F. Davenel, "FIDES Reliability Predictions," European Space Components Conference ESCCON 2016, Mar. 2016, Noordwijk, The Netherlands.
- [154] P. Pearson, R. Callen, J.-P. Blanquart, S. Bourbouse, and J.-F. Gajewski, "Reliability Prediction Data Sources and Methodologies for Space Applications RPDSM: Final Report," Aug. 2016.
- [155] S. Bourbouse *et al.*, "Evaluation of EEE & Mechanical reliability prediction models for space applications," in *A Workshop on Assessment of REliability - AWARE*, Sep. 2016.
- [156] W. Huang, J. Loman, R. Andrada, and R. Ortland, "A Bayesian posterior estimate of traveling wave tubes (TWT) failure rate based on spacecraft on-orbit flight data," Proceedings of the Annual Reliability and Maintainability Symposium (RAMS), Jan. 2016, Tucson, AZ, USA.
- [157] K. Hoefner, A. Vahl, and E. Stoll, "Satellite Architecture Optimization via Adapted Reliability Analyses from Commercial Aviation," Proceedings of the 68th International Astronautical Congress, 2017, Adelaide, Australia.
- [158] D. D. Dylis and M. G. Priore, "A comprehensive reliability assessment tool for electronic systems," in *Proceedings of the Annual Reliability and Maintainability Symposium 2001*, Jan. 2001, Philadelphia, PA, USA, pp. 308–313.
- [159] European Committee for Standardization, "Space engineering - Testing," European Standard EN 16603-10-03, Aug. 2014.
- [160] European Cooperation for Space Standardization (ECSS), "Space Engineering - Testing," ECSS-E-ST-10-03C, Jun. 2012.
- [161] International Organization for Standardization, "Space systems — General test methods for space craft, subsystems and units," ISO 15864:2004 (E), Aug. 2004.

- [162] European Cooperation for Space Standardization (ECSS), "Space engineering - Verification," ECSS-E-ST-10-02C, Mar. 2009.
- [163] European Cooperation for Space Standardization (ECSS), "Space product assurance - Components reliability data sources and their use," ECSS-Q-HB-30-08A, Jan. 2011.
- [164] European Cooperation for Space Standardization (ECSS), "Space engineering - Verification guidelines," ECSS-E-HB-10-02A, Dec. 2010.
- [165] Department of Defense, "Standard Practice - Product Verification Requirements for Launch, Upper Stage, and Space Vehicles," MIL-STD-1540D, Jan. 1999.
- [166] Department of Defense, "Design, Construction, and Testing Requirements for One of a Kind Space Equipment," Military Handbook, DOD-HDBK-343 (USAF), Feb. 1986.
- [167] NASA, "Payload Test Requirements," NASA Technical Standard, NASA-STD-7002A, Sep. 2004.
- [168] NASA Goddard Space Flight Center, "General Environmental Verification Standard (GEVS) for GSFC Flight Programs and Projects," GSFC-STD-7000A, Mar. 2013.
- [169] J. W. Welch, "Flight Unit Qualification Guidelines," Aerospace Report No. TOR-2010(8591)-20, Jun. 2010.
- [170] D. Parsley, "Space Mission Verification and Validation," in *Space mission engineering: The new SMAD*, J. R. Wertz, D. F. Everett, and J. J. Puschell, Eds., Hawthorne, CA: Microcosm Press, 2011, pp. 718–726.
- [171] L. A. Escobar and W. Q. Meeker, "A Review of Accelerated Test Models," *Statist. Sci.*, vol. 21, no. 4, pp. 552–577, 2006.
- [172] D. H. Collins, J. K. Freels, A. V. Huzurbazar, R. L. Warr, and B. P. Weaver, "Accelerated Test Methods for Reliability Prediction," *Journal of Quality Technology*, vol. 45, no. 3, pp. 244–259, 2017.
- [173] B. Kosinski and D. Cronin, "Highly Accelerated Life Test (HALT) Program at Space Systems Loral," 27th Aerospace Testing Seminar, Oct. 2012, Los Angeles, CA, USA.
- [174] W. Q. Meeker, G. Sarakakis, and A. Gerokostopoulos, "More Pitfalls of Accelerated Tests," *Journal of Quality Technology*, vol. 45, no. 3, pp. 213–222, 2017.
- [175] J. B. Hall, "Methodology for Evaluating Reliability Growth Programs of Discrete Systems," Dissertation, University of Maryland, College Park, 2008.
- [176] L. H. Crow, "Planning a Reliability Growth Program Utilizing Historical Data," Proceedings of the Annual Reliability and Maintainability Symposium (RAMS), Jan. 2011, Lake Buena Vista, FL, USA.
- [177] T. J. Duane, "Learning Curve Approach to Reliability Monitoring," *IEEE Transactions on Aerospace*, Volume 2, Number 2, Apr. 1964.
- [178] A. L. Goel and K. Okumoto, "Time-Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures," *IEEE Transactions on Reliability*, Vol. R-28, No.3, Aug. 1979.
- [179] M. Ohba, "Software reliability analysis models," *IBM J. Res. & Dev.*, vol. 28, no. 4, pp. 428–443, 1984.
- [180] S. Yamada and S. Osaki, "Software Reliability Growth Modeling: Models and Applications," *IEEE Trans. Software Eng.*, vol. SE-11, no. 12, pp. 1431–1437, 1985.
- [181] D. P. Gaver and P. A. Jacobs, "Reliability growth by failure mode removal," *Reliability Engineering & System Safety*, vol. 130, pp. 27–32, 2014.
- [182] R. Strunz and J. W. Herrmann, "Planning, tracking, and projecting reliability growth a Bayesian approach," in *Proceedings / Annual Reliability and Maintainability Symposium (RAMS)*, Jan. 2012, Reno, Nevada, USA, pp. 1–6.
- [183] J. W. Evans, M. P. Kaminsky, and L. D. Gallo Jr., "Reliability Growth Analysis of Satellite Systems," Annual Conference of the Prognostics and Health Management Society, Sep. 2012, Minneapolis, Minnesota, USA.

- [184] H. Sukhwani, J. Alonso, K. S. Trivedi, and I. Mcginnis, "Software Reliability Analysis of NASA Space Flight Software: A Practical Experience," 2016 IEEE International Conference on Software Quality, Reliability and Security (QRS), pp. 386–397, Aug. 2016, Vienna, Austria.
- [185] K. H. Cho, J. S. Jang, and S. C. Park, "NHPP Model based Reliability Growth Management of a Hybrid DC-DC Converter," International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 17 (2017) pp. 6919-6928.
- [186] H. Garrett, "Ultra-reliability and long-life," in *Space-based Infrared System Program*, Jet Propulsion Lab, Ed., 2002.
- [187] Department of Defense, "Test Method Standard for Microcircuits," MIL-STD-883E, Dec. 1996.
- [188] J. E. Tomayko, *Computers in Spaceflight: The NASA Experience*, NASA Contractor Report 182505, 1988.
- [189] McDonnell Douglas Astronautics Company - Engineering Services, "Independent Orbiter Assessment: Assessment of the Back-Up Flight System," Feb. 1988.
- [190] private communication of Steve Nagel (STS-55 Commander) to U. Walter (STS-55 payload astronaut).
- [191] G. J. Cancro and M. D. Trela, "STEREO Fault Protection Challenges and Lessons Learned," JOHNS HOPKINS APL TECHNICAL DIGEST, VOLUME 28, NUMBER 2, 2009.
- [192] A. Kleyner, "Reliability in the Automotive Industry – New Challenges and Solutions," Accelerated Stress Testing and Reliability Conference, Sep. 2016, Pensacola Beach, Florida, USA.
- [193] W. Molnau and J. Oliveri, "Multi-Spacecraft Manufacturing," in *Space mission engineering: The new SMAD*, J. R. Wertz, D. F. Everett, and J. J. Puschell, Eds., Hawthorne, CA: Microcosm Press, 2011, pp. 726–736.
- [194] M. Ahmed, "Test Like You Fly," Proceedings of 23rd Aerospace Testing Seminar 2006, Manhattan Beach, California, USA, Oct. 2006.
- [195] J. D. White, "Test Like You Fly: Assessment and Implementation Process," Aerospace Report TOR-2010(8591)-6, Jan. 2010.
- [196] M. Pasquinelli, P. Maggiore, S. Voglino, P. Messidoro, and V. Basso, "Evolution of Complexity Index Methodology for Spacecraft Cost Comparison and Prediction," Proceedings of 25th Aerospace Testing Seminar, Oct. 2008, Manhattan Beach, California, USA.
- [197] O. Brunner, H. Joumier, L. Montone, U. Ragnit, and M. Wagner, "The European Model- and Test Effectiveness Database MAT€D: Examples of Application," Proceedings of 24th Aerospace Testing Seminar, Apr. 2008, Manhattan Beach, California, USA.
- [198] T. Niwa, D. Takahashi, and Q. Shi, "Review JAXA Test Standard by the Lesson's Learned from Ground Test non-conformance database," Proceedings of 28th Aerospace Testing Seminar, Mar. 2014, Los Angeles, California, USA.
- [199] W. F. Tosney, "Knowledge Management Strategies for Space Program Development, Integration, and Test," Proceedings 4th International Symposium on Environmental Testing for Space Programmes, Liege, Belgium, Jun. 2001.
- [200] P. Messidoro *et al.*, "Status and Perspectives of the MAT€D Initiative on Test Perceptiveness and Effectiveness," Proceedings of 23rd Aerospace Testing Seminar 2006, Manhattan Beach, California, USA, Oct. 2006.
- [201] P. Maggiore, M. Pasquinelli, P. Messidoro, V. Basso, and G. Sembenini, "Comparative Analysis of Spacecraft Anomalies using Complexity Indexes: A Top-Down Approach," Proceedings of 24th Aerospace Testing Seminar 2008, Manhattan Beach, California, USA, Apr. 2008.
- [202] O. Brunner, "MATED - the User's Perspective," Proceedings of 26th Aerospace Testing Seminar, Mar. 2011, Los Angeles, California, USA.
- [203] P. Messidoro, M. Pasquinelli, V. Basso, and S. Voglino, "MATED Recent Project Feedback and Lessons Learned for the Next Challenges," Proceedings of 26th Aerospace Testing Seminar, Mar. 2011, Los Angeles, California, USA.

- [204] B. Laine *et al.*, “Analysis of Spacecraft qualification Sequence and Environmental Testing,” Proceedings of 29th Aerospace Testing Seminar, Oct. 2015, Los Angeles, California, USA.
- [205] M. Pasquinelli *et al.*, “Analysis of Spacecraft qualification Sequence and Environmental Testing: mid-term results,” Proceedings of 28th Aerospace Testing Seminar, Mar. 2014, Los Angeles, California, USA.
- [206] O. Brunner, A. Boerngen, and P. Messidoro, “MATED Software Updates and Latest Analysis Results,” Proceedings of 28th Aerospace Testing Seminar, Mar. 2014, Los Angeles, California, USA.
- [207] O. Brunner and D. Hagelschuer, “Model and Test Effectiveness Database MATED - Developments 2016-2017,” Proceedings of 30th Aerospace Testing Seminar, Mar. 2017, Los Angeles, California, USA.
- [208] J. H. Saleh, *Analyses for durability and system design lifetime: A multidisciplinary approach*. Cambridge, USA: Cambridge Univ. Press, 2008.
- [209] J. H. Saleh, J.-P. Torres-Padilla, D. E. Hastings, and D. J. Newman, “To Reduce or to Extend a Spacecraft Design Lifetime?,” *Journal of Spacecraft and Rockets*, vol. 43, no. 1, pp. 207–217, 2006.
- [210] C. Chaplin *et al.*, “NASA: Assessments of Selected Large-Scale Projects,” Report No. GAO-10-227SP, United States Government Accountability Office, 2010.
- [211] E. J. Tomei and I.-S. Chang, “51 Years of Space Launches and Failures,” IAC-09-D1.5.1, Proceedings of the 60th International Astronautical Congress, Daejeon, Republic of Korea, Oct. 2009.
- [212] S. W. Janson, “The History of Small Satellites,” in *Small satellites: Past, present, and future*, H. Helvajian, Ed., El Segundo Calif.: Aerospace Press, 2008.
- [213] Thomas D. Rivers, “Small Satellites - Evolving Innovation for the Entire Market,” in *31st Space Symposium*, Colorado Springs, Colorado, United States of America, 2015.
- [214] D. A. Bearden, “Small-Satellite Costs,” Crosslink Winter 2000/2001.
- [215] D. Ward, “Faster, Better, Cheaper Revisited: Program Management Lessons from NASA,” March-April 2010, Defense AT&L.
- [216] B. Horais, “Pioneering Innovation in Space - 30 Years of International Leadership,” *AIAA/USU Conference on Small Satellites, Year in Review*, SSC16-III-01, <https://digitalcommons.usu.edu/smallsat/2016/TS3YearInReview/1>, 2016.
- [217] T. D. Taverney, *Op-ed | Turning technology inside our adversaries*. SpaceNews, June 2017. [Online] Available: <http://spacenews.com/op-ed-turning-technology-inside-our-adversaries/>. Accessed on: Jan. 09 2018.
- [218] P. Swarts, *If America wants to succeed, it needs to learn to fail, top general says - SpaceNews.com*. [Online] Available: <http://spacenews.com/if-america-wants-to-succeed-it-needs-to-learn-to-fail-top-general-says/>. Accessed on: Feb. 27 2018.
- [219] G. F. Dubos and J. H. Saleh, “Risk of spacecraft on-orbit obsolescence: Novel framework, stochastic modeling, and implications,” *Acta Astronautica*, vol. 67, no. 1-2, pp. 155–172, 2010.
- [220] J. R. Werz, “What is Space Mission Engineering,” in *Space mission engineering: The new SMAD*, J. R. Wertz, D. F. Everett, and J. J. Puschell, Eds., Hawthorne, CA: Microcosm Press, 2011, pp. 1–4.
- [221] J. R. Wertz, R. C. Conger, M. Rufer, N. Sarzi-Amadé, and R. E. van Allen, “Methods for Achieving Dramatic Reductions in Space Mission Cost,” Reinventing Space Conference March 2–5, 2011 Los Angeles, California, USA.
- [222] California Polytechnic State University, “CubeSat Design Specification (CDS) REV 13,” 2014.
- [223] H. Heidt, J. Puig-Suari, A. S. Moore, S. Nakasuka, and R. J. Twiggs, “CubeSat: A new Generation of Picosatellite for Education and Industry Low-Cost Space Experimentation,” Proceedings of the AIAA/USU Conference on Small Satellites, Lessons Learned - In Success and Failure, SSC00-V-5, <https://digitalcommons.usu.edu/smallsat/2000/All2000/32/>.
- [224] SpaceWorks, “Nano/Microsatellite Market Forecast, 8th Edition,” 2018.

- [225] R. P. Welle, "The CubeSat Paradigm: An Evolutionary Approach to Satellite Design," 32nd Space Symposium, Apr. 2016, Colorado Springs, Colorado, USA.
- [226] Radius Space. [Online] Available: www.radiuspace.com. Accessed on: Oct. 10 2016.
- [227] Planetary Systems Corporation, "Canisterized Satellite Dispenser (CSD) Data Sheet: U.S. Patent 9,415,883 B2," Aug. 2017.
- [228] J. Straub, "Extending the Student Qualitative Undertaking Involvement Risk Model," *J. Aerosp. Technol. Manag.*, vol. 6, no. 3, pp. 333–352, 2014.
- [229] K. Woellert, "The Value of CubeSats to National Innovation Systems," IAFF258 Space Launch Tutorial, UN – ISU SSP 2011 TP Small Satellites TP2.
- [230] N. Jones, "Mini satellites prove their scientific power," *Nature*, vol. 508, no. 7496, pp. 300–301, 2014.
- [231] C. K. Pang, A. Kumar, C. H. Goh, and C. V. Le, "Nano-satellite swarm for SAR applications: Design and robust scheduling," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 2, pp. 853–865, 2015.
- [232] P. Gavigan, "Operational Use of Small Satellites for the Canadian Armed Forces," IAC-14.B4.4.7, Proceedings of the 65th International Astronautical Congress, Oct. 2014, Toronto, Canada.
- [233] Z. R. Manchester, "Centimeter-Scale Spacecraft: Design, Fabrication, and Deployment," Dissertation, Cornell University, 2015.
- [234] T. R. Perez and K. Subbarao, "A Survey of Current Femtosatellite Designs, Technologies, and Mission Concepts," *Journal of Small Satellites*, 2016, Vol.5, No.3, pp. 467-482.
- [235] K. Beckwith, "EEE Parts Database of CubeSat Projects and Kits," NASA Electronic Parts and Packaging (NEPP) Program Office of Safety and Mission Assurance, 2015.
- [236] K. Avery *et al.*, "Total Dose Test Results for CubeSat Electronics," in *IEEE Radiation Effects Data Workshop (REDW)*, Jul. 2011, Las Vegas, Nevada, USA, pp. 1–8.
- [237] W. Frazier, R. Rohrschneider, and M. Verzuh, "Cubesat Strategies for Long-Life Missions," 10th IAA Low-Cost Planetary Missions Conference, Jun. 2013, Pasadena, California, USA.
- [238] G. A. Johnson-Roth, "Key Considerations for Mission Success for Class C/D Mission," Aerospace Report No. TOR-2013-00294, Jun. 2013.
- [239] E. Deems, "Risk Management of Student-Run Small Satellite Programs," Proceedings of the 2006 AIAA/USU Conference on Small Satellites, Technical Session VII: University Programs, SSC06-VII-9, <https://digitalcommons.usu.edu/smallsat/2006/All2006/58/>.
- [240] J. Straub, R. Fevig, J. Casler, and O. Yadav, "Risk Analysis & Management in Student-Centered Spacecraft Development Projects," Proceedings of the Annual Reliability and Maintainability Symposium, Jan. 2013, Orlando, FL, USA.
- [241] J. Elstak, R. Amini, and R. Hamann, "A Comparative Analysis of Project Management and Systems Engineering Techniques in CubeSat Projects," *INCOSE International Symposium*, vol. 19, no. 1, pp. 545–559, 2009.
- [242] G. A. Johnson-Roth, "Mission Assurance Guidelines for A-D Mission Risk Classes," Aerospace Report No. TOR-2011(8591)-21, Jun. 2011.
- [243] K. M. Brumbaugh and E. G. Lightsey, "Application of Risk Management to University CubeSat Missions," *Journal of Small Satellites*, 2013, Vol. 2, No. 1, pp. 147-160.
- [244] M. Langer *et al.*, "Results and lessons learned from the CubeSat mission First-MOVE," in: *Small Satellite Missions for Earth Observation*, R. Sandau, H.-P. Roeser und A. Valenzuela, Springer Berlin Heidelberg, 2015.
- [245] K. H. Steinkirchner, "Project Management Risks in CubeSat Development," RT-MA 2017/23, Master's Thesis, Technical University of Munich, 2017.
- [246] M. Barschke and K. Gordon, "A Generic System Architecture for a Single-Failure Tolerant Nanosatellite Platform," IAC-14-B4.6A, Proceedings of the 65th International Astronautical Congress, Oct. 2014, Toronto, Canada.

- [247] M. Swartwout, "The First 272 CubeSats," EEE Parts for Small Missions 2014 Workshop, NASA Electronic Parts and Packaging Program (NEPP), NASA Goddard Space Flight Center, Sep. 2014.
- [248] M. Swartwout, "Secondary spacecraft in 2015: Analyzing success and failure," in *Proceedings of the IEEE Aerospace Conference*, Mar. 2015, Big Sky, Montana, USA.
- [249] M. Swartwout, "Secondary Spacecraft in 2016: Why Some Succeed (And Too Many Do Not)," *Proceedings of the 2016 IEEE Aerospace Conference*, Mar. 2016, Big Sky, Montana, USA.
- [250] M. Swartwout, "CubeSats and Mission Success: 2017 Update," 2017 Electronic Technology Workshop, NASA Electronic Parts and Packaging Program (NEPP), NASA Goddard Space Flight Center, Jun. 2017.
- [251] M. Swartwout, "University-Class Spacecraft by the Numbers: Success, Failure, Debris. (But Mostly Success.)," *Proceedings of the 30th Annual AIAA/USU Conference on Small Satellites*, Logan, UT, 6-11 August, 2016, Paper SSC16-XIII-1.
- [252] ESA-ESTEC, "Product and Quality Assurance Requirements for In-Orbit Demonstration CubeSat Projects," Mar. 2013.
- [253] P. Fiala and A. Vobornik, "Embedded microcontroller system for PilsenCUBE picosatellite," in *IEEE 16th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS)*, Apr. 2013, Karlovy Vary, Czech Republic, pp. 131–134.
- [254] G. Obiols Rabasa, "Methods for dependability analysis of small satellite missions," PhD Thesis, Politecnico di Torino, Jun. 2015.
- [255] G. Obiols Rabasa, "CubeSat Survey Results", Personal Communication, Sep. 2014.
- [256] M. Cho, "Monte Carlo Simulation of Reliability Growth of Small-Scale Satellites Through Testing," *Proceedings of the 65th International Astronautical Congress*, Oct. 2014, Toronto, Canada.
- [257] M. Cho and P. Faure, "Reliability Growth of a Nano-Satellite through Assembly, Integration and Testing," *Proceedings of the 66th International Astronautical Congress*, Oct. 2015, Jerusalem, Israel.
- [258] P. Faure, A. Tanaka, and M. Cho, "Toward the Improvement of Lean Satellites Reliability Through Testing – The HORYU-IV (AEGIS) Nano-Satellite Case Study," *Proceedings of the 67th International Astronautical Congress*, Sep. 2016, Guadalajara, Mexico.
- [259] P. Faure, A. Tanaka, and M. Cho, "Toward lean satellites reliability improvement using HORYU-IV project as case study," *Acta Astronautica*, vol. 133, pp. 33–49, 2017.
- [260] P. Faure, A. Tanaka, and M. Cho, "Multi-criteria Assessment for the Optimization of Lean Satellite Programs," *Proceedings of the 1st IAA Latin American Symposium on Small Satellites*, Mar. 2017, Buenos Aires, Argentina.
- [261] S. A. Jacklin, "Survey of Verification and Validation Techniques for Small Satellite Software Development," Space Tech Expo Conference, May. 2015, Long Beach, CA, USA.
- [262] F. Schummer, "Reliability Assessment of Small Spacecraft Before Launch," RT-MA 2017/22, Master's Thesis, Technical University of Munich, 2017.
- [263] ReliaSoft Corporation, "Life Data Analysis with Zero-Time (Out-Of-The-Box) Failures," *Reliability Hot Wire*, Issue 83, Jan. 2008.
- [264] M. Langer, M. Weisgerber, J. Bouwmeester, and A. Hoehn, "A reliability estimation tool for reducing infant mortality in CubeSat missions," in *2017 IEEE Aerospace Conference: Yellowstone Conference Center, Big Sky, Montana, March 4-11, 2017*, Big Sky, MT, USA, 2017, pp. 1–9.
- [265] M. Rupprecht, *DK3WN SatBlog*. [Online] Available: <http://www.dk3wn.info/satellites.shtml>. Accessed on: May 01 2018.
- [266] M. Wade, *Encyclopedia Astronautica*. [Online] Available: <http://www.astronautix.com/>. Accessed on: May 01 2018.
- [267] Technische Universität Berlin, *TUB Small Satellite Database*. [Online] Available: https://www.raumfahrttechnik.tu-berlin.de/%20menue/publikationen/small_satellite_database/. Accessed on: May 01 2018.

- [268] G. D. Krebs, *Gunter's Space Page*. [Online] Available: <http://space.skyrocket.de/>. Accessed on: May 01 2018.
- [269] E. Kulu, *NANOSATELLITE & CUBESAT DATABASE*. [Online] Available: <http://www.nanosats.eu/>. Accessed on: May 01 2018.
- [270] ESA, *Satellite Missions Database*. [Online] Available: <https://eoportal.org/web/eoportal/satellite-missions>. Accessed on: May 01 2018.
- [271] B. Klofas, J. Anderson, and K. Leveque, "A Survey of CubeSat Communication Systems," Proceedings of the 5th Annual CubeSat Developers' Workshop, 2008.
- [272] B. Klofas and K. Leveque, "A Survey of CubeSat Communication Systems: 2009–2012," Proceedings of the 10th Annual CubeSat Workshop. San Luis Obispo, California, Apr. 2013.
- [273] E. L. Kaplan and P. Meier, "Nonparametric Estimation from Incomplete Observations," *Journal of the American Statistical Association*, vol. 53, no. 282, p. 457, 1958.
- [274] M. Langer *et al.*, "Deployable Structures in the CubeSat Program MOVE," Proceedings of the 2nd International Conference "Advanced Lightweight Structures and Reflector Antennas", Oct. 2014, Sheraton Metechi Palace Hotel, Tbilisi, Georgia.
- [275] M. Langer *et al.*, "MOVE-II - The Munich Orbital Verification Experiment II," IAA-AAS-CU-17-06-05, Proceedings of the 4th IAA Conference on University Satellite Missions & CubeSat Workshop, Rome, Italy, Dec. 2017.
- [276] M. Langer *et al.*, "MOVE-II - der zweite Kleinsatellit der Technischen Universität München," Deutscher Luft- und Raumfahrtkongress 2015, Rostock, Germany, Sep. 2015.
- [277] J. Kiesbye, "Hardware-in-the-Loop Verification of the Distributed, Magnetorquer-Based Attitude Determination & Control System of the CubeSat MOVE-II," RT-MA 2017/16, Master's Thesis, Technical University of Munich, 2017.
- [278] A. Lill, "Organization and Development of the Mission Operations System for the MOVE-II CubeSat," RT-IDP 2016/05, Interdisciplinary Project, Technical University of Munich, 2018.
- [279] M. Grulich *et al.*, "SMARD-REXUS-18: Development and Verification of an SMA Based CubeSat Solar Panel Deployment Mechanism," Proc. 22nd ESA Symposium European Rocket & Balloon Programmes and Related Research, 2015, pp. 7–12.
- [280] M. Rutzinger *et al.*, "On-orbit verification of space solar cells on the CubeSat MOVE-II," in *2016 IEEE 43rd Photovoltaic Specialists Conference (PVSC): 5-10 June 2016*, Portland, OR, USA, 2016, pp. 2605–2609.
- [281] D. Messmann *et al.*, "Magnetic Attitude Control for the MOVE-II Mission," Proceedings of the 7th European Conference for Aeronautics and Space Sciences (EUCASS), Milan, Jun. 2017.
- [282] D. Messmann *et al.*, "Advances in the Development of the Attitude Determination and Control System of the CubeSat MOVE-II," Proceedings of the 7th European Conference for Aeronautics and Space Sciences (EUCASS), Milan, Jun. 2017.
- [283] N. Appel, S. Rueckerl, and M. Langer, "Nanolink: A robust and efficient protocol for Small Satellite radio links," Proceedings of the Small Satellites Systems and Services The 4S Symposium 2016, Valletta, Malta, 2016.
- [284] N. Appel *et al.*, "TDP-3 VANGUARD: Verification of a New Communication System ForCubeSats on BEXUS 22," Proceedings of the 23rd ESA Symposium on European Rocket and Balloon Programmes and Related Research, Visby, Sweden, 2017.
- [285] M. Langer, S. Rueckerl, and MOVE-II Team, *MOVE-II System Documentation*, Manuscript in preparation, 2018.
- [286] D. Voss *et al.*, "Educational Programs: Investment with a large return," Proceedings of the 26th Annual AIAA/USU Conference on Small Satellites, Paper SSC12-VII-1, Logan, UT, Aug. 2012, 2012.
- [287] J. T. Emison, K. Yoshino, S. E. Straits, and H. D. Voss, "Satellite Design for Undergraduate Senior Capstone," 121st ASEE National Conference, Jun. 2014, Indianapolis, IN, USA, 2014.

- [288] M. Langer, *Munich Orbital Verification Experiment II*. [Online] Available: www.move2space.de. Accessed on: May 05 2018.
- [289] J. A. Morton, "The Role of Components in Satellite Reliability," *IRE TRANSACTIONS ON RELIABILITY AND QUALITY CONTROL*, Jul. 1962.
- [290] Z. Jiang, K. P. Scheibe, S. Nilakanta, and X. S. Qu, "The Economics of Public Beta Testing," *Decision Sciences*, vol. 48, no. 1, pp. 150–175, 2017.
- [291] M. D. Trela and J. Maximoff, "8.3.2 Practical Approach to Increasing State Space Coverage for System Testing," *INCOSE International Symposium*, vol. 20, no. 1, pp. 1062–1072, 2010.
- [292] D. R. Kuhn, D. R. Wallace, and A. M. Gallo, "Software fault interactions and implications for software testing," *IEEE Trans. Software Eng.*, vol. 30, no. 6, pp. 418–421, 2004.
- [293] Jean-Philippe Lang, *Redmine 3.1.7.stable*, www.redmine.org, 2016.
- [294] S. Butterfield, E. Costello, C. Henderson, and S. Mourachov, *Slack*: Slack Technologies, <https://slack.com>, 2018.
- [295] S. Yamada, M. Ohba, and S. Osaki, "S-Shaped Reliability Growth Modeling for Software Error Detection," *IEEE Trans. Rel.*, vol. R-32, no. 5, pp. 475–484, 1983.
- [296] M. Weisgerber, "Reliability Prediction of Student-Built CubeSats in Early Project Phases," RT-MA 2017/26, Master's Thesis, Technical University of Munich, 2018.
- [297] ALD Services, *Free MTBF Calculator*. [Online] Available: <https://aldservice.com/Reliability-Software/free-mtbf-calculator.html>. Accessed on: May 13 2018.
- [298] Reliability Data of the MSP430FR2433 predicted using the DPPM/FIT/MTBF estimator of Texas Instruments, *Texas Instruments*. [Online] Available: <http://www.ti.com/quality/docs/estimator.tsp>. Accessed on: May 08 2018.
- [299] S. Minderhoud and P. Fraser, "Shifting paradigms of product development in fast and dynamic markets," *Reliability Engineering & System Safety*, vol. 88, no. 2, pp. 127–135, 2005.
- [300] L. J. Paxton, "'Faster, better, and cheaper' at NASA: Lessons learned in managing and accepting risk," *Acta Astronautica*, vol. 61, no. 10, pp. 954–963, 2007.
- [301] German Federal Bureau of Aircraft Accident Investigation, *Investigation Reports*. [Online] Available: https://www.bfu-web.de/EN/Publications/Investigation%20Report/reports_node.html. Accessed on: May 14 2018.
- [302] D. Sinclair and J. Dyer, "Radiation Effects and COTS Parts in SmallSats," 27th Annual AIAA/UAA Conference on Small Satellites, 2013, Logan, Utah, USA.
- [303] R. Kingsbury, F. Schmidt, K. Cahoy, and D. Sklair, "TID Tolerance of Popular CubeSat Components," Proceedings of the 50th Nuclear and Space Radiation Effects Conference (NSREC) 2013.
- [304] S. Gerardin *et al.*, "Radiation Effects in Flash Memories," *IEEE Trans. Nucl. Sci.*, vol. 60, no. 3, pp. 1953–1969, 2013.
- [305] P. B. de Selding, *1 in 5 Cubesats Violates International Orbit Disposal Guidelines*. [Online] Available: <http://spacenews.com/1-in-5-cubesats-violate-international-orbit-disposal-guidelines/>. Accessed on: May 08 2018.
- [306] M. Swartwout, "CubeSats: Toys, Tools or Debris Cloud?," St. Louis Space Frontier Gateway to Space Conference, Nov. 2014.
- [307] H. Krag, "The Inter-Agency Space Debris Coordination Committee: An overview of the IADC annual activities," 54th Session of the Scientific and Technical Subcommittee, United Nations Committee on the Peaceful Uses of Outer Space, Feb. 2017.
- [308] J. Radtke, E. Stoll, H. Lewis, and B. Bastida Virgili, "The Impact of the Increase in Small Satellite Launch Traffic on the Long-Term Evolution of the Space Debris Environment," 7th European Conference on Space Debris, Apr. 2017, Darmstadt, Germany.

- [309] J. Gutmiedl *et al.*, “Providing Hands-On Space Education by Involvement of Collaborating Self-Reliant Student Teams,” 1st Symposium on Space Educational Activities, ESA, Padua, Italy, Dec. 2015.
- [310] D. D. Galorath and M. W. Evans, *Software sizing, estimation, and risk management: When performance is measured performance improves*. Boca Raton, FL: Auerbach Publications, 2006.
- [311] N. J. Lindsey, N. Rackley, A. Brall, and A. Mosleh, “Reliability Prediction Using Bayesian Updating of On-Orbit Performance,” Annual Reliability and Maintainability Symposium (RAMS); Jan. 2013; Orlando, FL; United States.
- [312] L. H. Crow, “Evaluating the reliability of repairable systems,” Proceedings of the Annual Reliability and Maintainability Symposium (RAMS), 1990.
- [313] Y. K. Malaiya, “Antirandom testing: Getting the most out of black-box testing,” in *Proceedings: The Sixth International Symposium on Software Reliability Engineering*, Toulouse, France, Oct. 1995, pp. 86–95.
- [314] S. Nidhra and J. Dondeti, “Black Box and White Box Testing Techniques - A Literature Review,” *IJESA*, vol. 2, no. 2, pp. 29–50, 2012.
- [315] K. Wortman, B. Duncan, and E. Melin, *Agile Methodology for Spacecraft Ground Software Development: A Cultural Shift*, IEEE Aerospace Conference, 2017, Big Sky, Montana, USA.
- [316] P. C. Mehlitz and J. Penix, “Expecting the Unexpected: Radiation Hardened Software,” Infotech@Aerospace, AIAA 2005, Arlington VA, Sep. 2005.
- [317] S. Engelen, E. Gill, and C. Verhoeven, “On the reliability, availability, and throughput of satellite swarms,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 2, pp. 1027–1037, 2014.
- [318] R. Lai and M. Garg, “A Detailed Study of NHPP Software Reliability Models (Invited Paper),” *JSW*, vol. 7, no. 6, 2012.
- [319] V. Costabile, A. Discepoli, C. Fiorentino, and G. Morelli, “New spacecraft assembly, integration and test approach for Globalstar satellite constellation,” in *16th International Communications Satellite Systems Conference*, Washington, DC, USA, 1996.

List of Publications

- [1] A. Lill, T. Zwickl, C. Costescu, L. Patzwahl, C. Soare, and **M. Langer**, “Agile Mission Operations in the CubeSat Project MOVE-II“, AIAA 2018-2635, 2018 SpaceOps Conference, May 28-June 1, 2018, Marseille, France. <https://doi.org/10.2514/6.2018-2635>
- [2] **M. Langer**, F. Schummer, N. Appel, T. Gruebler, K. Janzer, J. Kiesbye, L. Krempel, A. Lill, D. Messmann, S. Rueckerl, M. Weisgerber, “MOVE-II - The Munich Orbital Verification Experiment II”, IAA-AAS-CU-17-06-05, Proceedings of the 4th IAA Conference on University Satellite Missions & CubeSat Workshop, Rome, Italy, December 2017.
- [3] T. Sinn, N. Reichenbach, M. Schimmerohn, C. Horch, **M. Langer**, P. Seefeldt, “The development of a passive de-orbit subsystem for small and micro satellites”, IAC-17,B4,6A,5,x39040, Proceedings of the 68th International Astronautical Congress, Adelaide, Australia, September 2017.
- [4] M. Weisgerber, F. Schummer, K. Steinkirchner, **M. Langer**, “Risk Reduction and Process Acceleration for Small Spacecraft Assembly and Testing by Rapid Prototyping”, Deutscher Luft- und Raumfahrtkongress 2017, Munich, Germany, September 5-8, 2017.
- [5] A. Lill, D. Messmann, **M. Langer**, “Agile Software Development for Space Applications”, Deutscher Luft- und Raumfahrtkongress 2017, Munich, Germany, September 5-8, 2017.

-
- [6] A. Hein, K. F. Long, D. Fries, N. Perakis, A. Genovese, S. Zeidler, **M. Langer**, R. Osborne, R. Swinney, J. Davies, B. Cress, M. Casson, A. Mann, R. Armstrong, "The Andromeda Study: A Femto-Spacecraft Mission to Alpha Centauri" arXiv preprint arXiv:1708.03556.
- [7] K. Antonini, **M. Langer**, A. Farid, U. Walter, "SWEET CubeSat – Water detection and water quality monitoring for the 21st century", In Acta Astronautica, Volume 140, 2017, Pages 10-17, ISSN 0094-5765, <https://doi.org/10.1016/j.actaastro.2017.07.046>.
- [8] D. Messmann, T. Gruebler, F. Coelho, T. Ohlenforst, J. van Bruegge, F. Mauracher, M. Doetterl, S. Plamauer, P. Schnierle, T. Kale, M. Seifert, A. Fuhrmann, E. Karagiannis, A. Ulanowski, T. Lausenhammer, A. Meraner, **M. Langer**, "Advances in the Development of the Attitude Determination and Control System of the CubeSat MOVE-II", 7th European Conference for Aeronautics and Space Sciences (EUCASS), Milan, Italy, June 3 - June 6, 2017. DOI: 10.13009/EUCASS2017-660
- [9] D. Messmann, F. Coelho, P. Niermeyer, **M. Langer**, H. Huang, U. Walter, "Magnetic Attitude Control for the MOVE-II Mission", 7th European Conference for Aeronautics and Space Sciences (EUCASS), Milan, Italy, June 3 - June 6, 2017. DOI: 10.13009/EUCASS2017-664
- [10] N. Appel, A. Kimpe, K. Kraus, **M. Langer**, M. Losekamm, M. Milde, T. Pöschl, S. Ruckerl, F. Schäfer, A. Stromsky, K. Würfl, "TDP-3 Vanguard: Verification of a New Communication System for CubeSats on BEXUS 22", Proc. 23rd ESA Symposium European Rocket & Balloon Programs and Related Research', 11–15 June 2017, Visby, Sweden.
- [11] S. Plamauer and **M. Langer**, "Evaluation of MicroPython as Application Layer Programming Language on CubeSats." ARCS 2017; 30th International Conference on Architecture of Computing Systems; Proceedings of, VDE, 2017. ISBN 978-3-8007-4395-7
- [12] **M. Langer**, M. Weisgerber, J. Bouwmeester, A. Hoehn, "A Reliability Estimation Tool for Reducing Infant Mortality in CubeSat Missions", proceedings of the IEEE Aerospace Conference, Mar 4 - 11, 2017, Big Sky, Montana, USA. DOI: 10.1109/AERO.2017.7943598
- [13] A. Soni, C. Welch, **M. Langer**, "Minimum Interstellar Precursor Mission", Proceedings of the 67th International Astronautical Congress, Guadalajara, Mexico, September 2016.
- [14] K. Antonini, **M. Langer**, A. Farid, U. Walter, "SWEET CubeSat – Water Detection and Water Quality Monitoring for the 21st Century", Proceedings of the 67th International Astronautical Congress, Guadalajara, Mexico, September 2016.
- [15] J. Bouwmeester, **M. Langer**, E. Gill, "Survey on the implementation and reliability of CubeSat electrical bus interfaces." CEAS Space Journal (2016): 1-11
- [16] **M. Langer**, J. Bouwmeester, "Reliability of CubeSats – Statistical Data, Developers' Beliefs and the Way Forward", Proceedings of the 30th Annual AIAA/USU Conference on Small Satellites, Logan, UT, 6-11 August, 2016, Paper SSC16-X-2.
- [17] M. Rutzinger, L. Krempel, M. Salzberger, M. Buchner, A. Höhn, M. Kellner, K. Janzer, C. G. Zimmermann, **M. Langer**, "On-Orbit Verification of Space Solar Cells on the CubeSat MOVE-II", 43rd Photovoltaic Specialist Conference (PVSC), IEEE, Portland, Oregon, June 5- June 10, 2016.
- [18] C. Fuchs, N. Dafinger, **M. Langer**, C. Trinitis, "Enhancing Nanosatellite Dependability Through Autonomous Chip-Level Debug Capabilities", Proceedings of the Small Satellites Systems and Services – The 4S Symposium 2016, Valletta, Malta, May 30–June 3, 2016.
- [19] N. Appel, S. Ruckerl, **M. Langer**, "Nanolink: A Robust and Efficient Protocol for Small Satellite Radio Links", Proceedings of the Small Satellites Systems and Services – The 4S Symposium 2016, Valletta, Malta, May 30–June 3, 2016.
-

- [20] C. M. Fuchs, N. Dafinger, **M. Langer** and C. Trinitis, "Enhancing Nanosatellite Dependability Through Autonomous Chip-Level Debug Capabilities," ARCS 2016; Proceedings of the 29th International Conference on Architecture of Computing Systems, Nuremberg, Germany, 2016, ISBN: 978-3-8007-4157-1, pp. 1-4.
- [21] J. Gutmiedl, M. Dziura, **M. Langer**, M. Losekamm, M. Grulich, M. Mutschler, "Providing Hands-On Space Education by Involvement of Collaborating Self-Reliant Student Teams", in: 1st Symposium on Space Educational Activities, ESA, Padua, Italy, 9-12 December 2015.
- [22] **M. Langer**, N. Appel, M. Dziura, C. Fuchs, P. Günzel, J. Gutmiedl, M. Losekamm, D. Meßmann, C. Trinitis, "MOVE-II – der zweite Kleinsatellit der Technischen Universität München", Deutscher Luft- und Raumfahrtkongress 2015, Rostock, Germany, September 22-24 2015.
- [23] C. Fuchs, **M. Langer** and C. Trinitis, "Enabling Dependable Data Storage for Miniaturized Satellites", in: Proceedings of the 29th AIAA/USU Conference on Small Satellites, Student Competition, SSC15-VIII-6. <http://digitalcommons.usu.edu/smallsat/2015/all2015/59/>.
- [24] C. Fuchs, **M. Langer** and C. Trinitis, "A Fault-Tolerant Radiation-Robust Filesystem for Space Use", In: Pinho L., Karl W., Cohen A., Brinkschulte U. (eds) Architecture of Computing Systems – ARCS 2015. ARCS 2015. Lecture Notes in Computer Science, vol 9017. Springer, Cham.
- [25] **M. Langer**, C. Olthoff, J. Harder, C. Fuchs, M. Dziura, A. Hoehn, U. Walter, "Results and lessons learned from the CubeSat mission First-MOVE", in: Small Satellite Missions for Earth Observation, R. Sandau, H.-P. Roeser und A. Valenzuela, Springer Berlin Heidelberg, 2015.
- [26] P. Zimmerhagl, J. Fagerudd, N. Ivchenko, G. Tibert, **M. Langer** and SEAM Team, "The Structural and Thermal Design Analysis of the SEAM CubeSat", in: Small Satellite Missions for Earth Observation, R. Sandau, H.-P. Roeser und A. Valenzuela, Springer Berlin Heidelberg, 2015.
- [27] C. Fuchs, **M. Langer**, C. Trinitis and N. Appel, "Dependable Computing for Miniaturized Satellites", 4th EIROforum School on Instrumentation (ESI 2015), ESO & EuroFusion, Garching, Germany, June 15-19, 2015.
- [28] C. Fuchs, **M. Langer**, and C. Trinitis, "A Fault-Tolerant Radiation-Robust Mass Storage Concept for Highly Scaled Flash Memory", in: DASIA 2015 - Data Systems in Aerospace, Barcelona, Spain 19-21 May, 2015, ESA-SP Vol. 732, 2015, id.13.
- [29] C. Fuchs, **M. Langer** and C. Trinitis, "Towards Fault-Tolerant Radiation-Robust Data Storage through Software Measures", in: The 8th ESA Workshop on Avionics Data, Control and Software Systems, European Space Research and Technology Centre (ESTEC), Leiden, The Netherlands, October 27-29, 2014.
- [30] **M. Langer**, C. Olthoff, L. Datshvili, H. Baier, N. Maghaldadze, U. Walter, "Deployable Structures in the CubeSat Program MOVE", Proceedings of the 2nd International Conference Advanced Lightweight Structures and Reflector Antennas, pp. 224-233, Tbilisi, Georgia, 1-3 October 2014.
- [31] M. Deiml, M. Kaufmann, P. Knieling, F. Olschewski, P. Toumpas, **M. Langer**, M. Ern, R. Koppmann, M. Riese, „DiSSECT – Development of a small satellite for climate research“, Proceedings of the 65th International Astronautical Congress, Toronto, Canada, October 2014.
- [32] M. J. Losekamm, T. Pöschl, **M. Langer**, S. Paul, „The AFIS detector: measuring antimatter fluxes on Nanosatellites“, Proceedings of the 65th International Astronautical Congress, Toronto, Canada, October 2014.

List of Supervised Theses

Master's Theses & Equivalentents

- [1] M. Dziura, "Development of a Software Independent Solution for Memory Integrity Testing on the Nanosatellite Mission MOVE 2", Diploma Thesis, TUM, 2014.
- [2] A. Abdellatif, "Feasibility Analysis of the Small Satellite Mission "ALTAIR"", Master's Thesis, TUM, 2014.
- [3] P. Zimmerhagl, "Development, Testing and Mechanical Analysis of a CubeSat Structure", Master's Thesis, TUM + KTH Stockholm, 2015.
- [4] C. Fuchs, "Dependable Computer Architectures and Software Concepts for Next-Generation Nanosatellites", Master's Thesis, TUM, 2015.
Awarded 2nd price in Frank J. Redd Student Competition, 29th AIAA/USU Conference on Small Satellite, US\$7,500
Awarded 1st price of ZARM Student Competition 2016, US\$1,300.
- [5] M. Burkart, "Integration and Test of the MiniRad Radiometer of the Satellite PolarCube", Diploma Thesis, TUM + Colorado Space Grant, 2015.
- [6] K. Antonini, "Requirements Development for the SWEET CubeSat", Master's Thesis, TUM, 2016.
- [7] S. Scheiblauer, "W-Band Antenna Design for CubeSat Applications", Master's Thesis, TUM, 2017.
- [8] S. Ruckerl, "Development and integration of a communication module for Nanosatellites based on VHDL within the MOVE-II mission", Master's Thesis, TUM, 2017.
Awarded 1st price of ZARM Student Competition 2017, US\$1,200.
- [9] S. Plamauer, "Evaluation of MicroPython as Application Layer Programming Language on Small Satellites", Master's Thesis, TUM, 2017.
- [10] N. Appel, "Development and Evaluation of Radio Transceivers for the CubeSat MOVE-II", Master's Thesis, TUM, 2017.
- [11] T. Grüberl, "Highly Integrated Smart Satellite Panels for Commercial Space Applications", Master's Thesis, TUM, 2017.
- [12] J. Kiesbye, "Hardware-in-the-loop verification of the distributed, magnetorquer based attitude determination & control system for the MOVE-II CubeSat", Master's Thesis, TUM, 2017.
- [13] K. Steinkirchner, "Project Management Risks in CubeSat Development", Master's Thesis, TUM, 2017.
- [14] F. Schummer, "Robustness Analysis on Small Spacecraft Design", Master's Thesis, TUM, 2017.
- [15] N. Reichenbach, "Development of the De-Orbit Subsystem for the 12U-CubeSat ERNST", Master's Thesis, TUM, 2018.
- [16] M. Weisgerber, "Reliability Assessment of CubeSats", Master's Thesis, TUM, 2018.
- [17] D. Vogel, "Combining Bio TRIZ and Additive Manufacturing for Satellite Structure Optimization", Master's Thesis, TUM, 2018.
- [18] D. Messmann, "Attitude Estimation on the CubeSat MOVE-II", Master's Thesis, TUM, ongoing.

- [19] A. Lill, "Design and Evaluation of an Agile Software Development Process for Space Applications", Master's Thesis, TUM, ongoing.

Bachelor's Theses & Equivalentents

- [20] M. Dziura, "Preparation and Conception of a CDH-System for the Nanosatellite MOVE-II", Semester Thesis, TUM, 2013.
- [21] C. Fuchs, "Evaluation and Preparation of an Operating System for the Nanosatellite Mission MOVE II", Interdisciplinary Project, TUM, 2014.
- [22] L. Schrenk, "Evaluation of highly foldable gravity gradient booms for the CubeSat Mission MOVE-II", Semester Thesis, TUM, 2014.
- [23] D. Messmann, "Development of a Simulation Environment with the STK/MATLAB Interface for Attitude Determination and Control of the Nanosatellite MOVE II", Semester Thesis, TUM, 2014.
- [24] J. Kiesbye, "Development of a Transceiver Prototype for a Nanosatellite Mission", Bachelor's Thesis, TUM, 2015.
- [25] N. Dafinger, "Designing a JTAG Communication Module for the Nanosatellite Mission MOVE-II", Interdisciplinary Project, TUM, 2015.
- [26] M. Stefan, "Finite Element Method (FEM) based Verification of the Vibration and Acceleration Test Data of the Nanosatellite First-Move", Semester Thesis, TUM, 2015.
- [27] M. Burkhart, "Testbed for CubeSat Attitude Determination Systems", Semester Thesis, TUM, 2015.
- [28] T. Assmann, "Development of a Solar Panel Deployment System for the Nanosatellite MOVE-II", Bachelor's Thesis, TUM, 2015.
- [29] D. Vogel, "Development of an Antenna Deployment Mechanism for the Nanosatellite MOVE-II", Bachelor's Thesis, TUM, 2015.
- [30] A. Wiedfeld, "Determination and Simulation of a Solar Panel Deployment Strategy for the Nanosatellite MOVE-II", Semester Thesis, TUM, 2015.
- [31] M. Gürster, "Hardware Setup of a Single-Ion Paul Trap for an Optical Atomic Clock for Space Applications", Semester Thesis, TUM, 2015.
- [32] D. Dorezyuk, "Design and implementation of a payload structure for a solar cell experiment on the High Altitude Pseudo Satellite (HAPS)", Semester Thesis, TUM, 2015.
- [33] S. Barth, "Design and Development of a system on a chip (SoC) for a CubeSat", Interdisciplinary Project, TUM, 2016.
- [34] R. Setter, "Development and Test of the Antenna Deployment Mechanism for the Nanosatellite MOVE-II, Bachelor's Thesis, TUM, 2016.
- [35] A. Feigel, "Design and validation of an infrared camera test setup inside a thermal vacuum chamber", Bachelor's Thesis, TUM, 2016.
- [36] N. Appel and S. Rueckerl, "Software for the MOVE-II Transceivers", Interdisciplinary Project, TUM, 2016.
- [37] J. Sticha, "Integrate Linux based Operating System in MOVE-II's On-Board Computer", Interdisciplinary Project, TUM, 2016.

-
- [38] M. Weisgerber, "Reliability Modeling Optimization for CubeSats", Semester Thesis, TUM, 2016.
 - [39] K. Janzer, "Thermal Analysis and Test of Components for the CubeSat MOVE-II", Semester Thesis, TUM, 2016.
 - [40] M. Baade, "Structural and Vibrational Analysis of the CubeSat MOVE-II", Semester Thesis, TUM, 2016.
 - [41] N. Wetter, "Development of an On-Orbit Simulation Environment for CubeSats", Bachelor's Thesis, TUM, 2016.
 - [42] J. Jelten and A. Kupka, "High Level Communication System for Data Transmission between the MOVE-II CubeSat and its Ground Station", Interdisciplinary Project, TUM, 2017.
 - [43] T. Seyidov, "Management and Scheduling of the Attitude Estimation & Control System Operation Modes for the MOVE-II Satellite", Interdisciplinary Project, TUM, 2017.
 - [44] A. Obada, "Implementation of a Ground Station Software for the Satellite Mission MOVE-II", Interdisciplinary Project, TUM, 2017.
 - [45] D. Messmann, "Development of a Sun Pointing Attitude Controller using Magnetic Actuation for the MOVE-II CubeSat Mission", Semester Thesis, TUM, 2017.
 - [46] E. Bayraktar, "File-Transfer and Executor Tools for MOVE-II", Interdisciplinary Project, TUM, 2017.
 - [47] E. Tarakci, "Fault Diagnostics and Health Monitoring for the Satellite Mission MOVE-II", Interdisciplinary Project, TUM, 2017.
 - [48] L. Krempel, "Device Development for Organic Solar Cell Characterization on Near Space Missions", Bachelor's Thesis, TUM, 2017.
 - [49] K. Wuerl, "Digital Modulation and Transmission Techniques for the MOVE-II Satellite-Ground-Station", Interdisciplinary Project, TUM, 2017.
 - [50] F. Huseynli, "Implementation of a Rescue Tool for the Satellite Mission MOVE-II", Interdisciplinary Project, TUM, 2017.
 - [51] N. Frinker, "Implementation of a Housekeeping and Data Controller for the Satellite Mission MOVE-II", Interdisciplinary Project, TUM, 2017.
 - [52] A. Fuhrmann, "On-Orbit Updates of the Attitude Estimation and Control Software for the MOVE-II Satellite", Interdisciplinary Project, TUM, 2017.
 - [53] P. Lux, "Space Debris Analysis and Space Debris Mitigation of CubeSats", Bachelor's Thesis, TUM, 2017.
 - [54] L. Geismayr, "Evaluation and Quantitative Characterization of Additive Manufacturing Processes for Spaceflight Applications", Bachelor's Thesis, TUM, 2017.
 - [55] J. Völker, "Design of an Optical Camera System Module for the TOM-Mission", Semester Thesis, TUM, 2017.
 - [56] S. Köchel, "New Space: Impacts of Future Concepts in Satellite Production on the Space Industry", Semester Thesis, TUM, 2017.
 - [57] A. Lill, "Organization and Development of the Mission Operations System for the MOVE-II CubeSat", Interdisciplinary Project, TUM, 2018.
 - [58] M. Doetterl, "Development of the ADCS Software", Interdisciplinary Project, TUM, 2018.
 - [59] F. Mauracher, "Development of the ADCS Software", Interdisciplinary Project, TUM, 2018.

- [60] L. Donini, "Implementation of a Packet-Based Protocol for the Satellite Mission MOVE-II", Interdisciplinary Project, TUM, 2018.
- [61] S. Drugalev, "Implementation of a Daemon for Critical Subsystems of the Satellite Mission MOVE-II", Interdisciplinary Project, TUM, 2018.
- [62] T. Kale, "Optimization and Verification of the Attitude Estimation & control Software for the MOVE-II Satellite", Interdisciplinary Project, TUM, ongoing.
- [63] W. Zimmer, "Implementation of a User-Interface for the Mission Control Centre for the Satellite Mission MOVE-II", Interdisciplinary Project, TUM, ongoing.
- [64] Marcel Cordovi, "Implementing a Kalman Filter for the MOVE-II ADCS", Interdisciplinary Project, TUM, ongoing.

Appendix A List of Figures and Tables

A.1 List of Figures

Figure 2-1: Electronic Parts Count in Space Missions over Time	9
Figure 2-2: Number $v(t)$ of non-repairable items still working at time t	11
Figure 2-3: Effects of different shape factors & scale factors.	13
Figure 2-4: The bathtub curve with infant mortality (1.) constant failure rate (2.) and wear-out (3.)	14
Figure 2-5: Failure Mechanisms in Spacecraft	18
Figure 2-6: Exponential SLOC growth in spaceflight projects.....	19
Figure 2-7: Breakdown of the causes of spacecraft faults analyzed by the Aerospace Company	21
Figure 2-8: Breakdown of causes for anomalies on satellites by EADS Astrium (2012)	22
Figure 2-9: Saturation curve system level testing of seven flight model spacecraft	25
Figure 2-10: First month and total life on-orbit malfunctions of 57 spacecraft	26
Figure 2-11: Number of spacecraft alive at a certain day after launch.....	27
Figure 2-12: Saturation curve of failures vs. test days of 39 spacecraft flight models	28
Figure 2-13: Weibull growth curve of on orbit failures and saturation curves of MTBF.	28
Figure 2-14: Component reliability of 57 GSFC missions.....	29
Figure 2-15: Growth of mass, power and parts count of early communication satellites.....	29
Figure 2-16: Parametric and non-parametric reliability of 1970–1980 spacecraft	30
Figure 2-17: Weibull fit of on-orbit failure data of 300 satellites	31
Figure 2-18: Predicted and experienced failure rate of the Voyager spacecraft.....	32
Figure 2-19: Anomalies of seven Mars missions from 1990 to 2004.	33
Figure 2-20: Non-parametric and parametric reliability of Pico- and Nanosatellites	35
Figure 2-21: Non-parametric (left) and parametric (right) reliability of the satellites.....	36
Figure 2-22: Non-parametric reliability and parametric 2-Weibull function fit of small satellite reliability. .	37
Figure 2-23: Non-parametric satellite reliability of LEO and GEO satellites	38
Figure 2-24: 2-Weibull mixture fit of satellite reliability	38
Figure 2-25: The General Shape of the Cost-Reliability Curve.....	41
Figure 2-26: System-level test sequence for spacecraft	48
Figure 2-27: The V-Modell of development and verification of space hardware.....	49
Figure 2-28: Highly accelerated tests used to identify weaknesses of different product phases.....	50
Figure 2-29: Reliability growth experienced in different systems under test.....	52
Figure 2-30: Reliability prediction and reliability growth for a serial-produced, commercial item	53
Figure 2-31: Delayed S-Shape growth model and testing effort of the same project over time.....	54
Figure 2-32: Reliability growth of the series of TRDS satellites and GOES satellites.....	55
Figure 2-33: Cumulative number of failures of the first generation GOES satellites.	56
Figure 2-34: Cumulative number of failures of the second generation of satellites of GOES satellites.....	56
Figure 2-35: Anomalies found in releases of on-board software for a NASA mission	56
Figure 2-36: Reliability of k-out-of-n redundancies and system vs. partitioned redundancy.	59
Figure 2-37: Historical failure rate of unscreened, Class B and Class S screened parts.....	60
Figure 2-38: Probability of a latent problem in a new design and in mass-produced components	62
Figure 2-39: Correlation of anomalies in ground testing (NCR) to flight anomalies (FA)	64
Figure 2-40: Cause of major and critical anomalies on spacecraft during functional tests	64
Figure 2-41: Launch mass of the first 30 spacecraft and the masses of Explorer spacecraft.....	66

Figure 2-42: Successful, failed and impaired missions analyzed by the Aerospace Company	67
Figure 2-43: GSD, Data Rate and Data Volume increase of SmallSats.....	68
Figure 2-44: Traditional Space Spiral (left) and the potential of SmallSats to reverse this spiral (right)	69
Figure 2-45: Standardized CubeSat sizes.....	70
Figure 2-46: Risk Surface of different classes of NASA missions	73
Figure 2-47: Success rate of all CubeSats launched	77
Figure 2-49: HORYU-IV cumulative failure rate and resulting Duane plot.....	78
Figure 4-1: Satellites failed in the first 7 years due to the 2-Weibull function of Castet & Saleh	86
Figure 4-2: Satellites failed in the first 14.5 years due to the 2-Weibull function of Castet & Saleh.	86
Figure 4-3: Satellites failed due to the infant mortality term of Castet & Saleh	87
Figure 4-4: Satellites failed due to the wear-out term of Castet & Saleh	87
Figure 4-5: Satellites failed in the first 7 years due to the 2-Weibull function of Saleh & Castet	88
Figure 4-6: Satellites failed in the first 14.5 years due to the 2-Weibull function of Saleh & Castet	89
Figure 4-7: Fraction failed due to the infant mortality term of Saleh & Castet.....	90
Figure 4-8: Fraction failed due to the wear-out term of Saleh & Castet.....	90
Figure 4-9: Modified Weibull extension fit with a bathtub-shaped failure rate function	91
Figure 4-10: Weibull-plot and Box-plot of the residuals for modified Weibull function	92
Figure 4-11: Modified Weibull fit vs. 2-Weibull fits of Castet & Saleh and Saleh & Castet	92
Figure 4-12: Difference of the 2-Weibull mixture fit to Castet & Saleh and Saleh & Castet.....	93
Figure 4-13: Failure rate of the modified Weibull function.....	94
Figure 4-14: Modified Weibull fit vs. modified Weibull fit by Peng & Zhang	95
Figure 4-15: 2-Weibull mixture fit and underlying nonparametric estimation	95
Figure 4-16: Satellites failed in the first 14.5 years due to the new 2-Weibull mixture function.	96
Figure 4-17: Satellites failed due to the infant mortality term of the new 2-Weibull mixture function.....	96
Figure 4-18: Weibull-plot and Box-plot of the new 2-Weibull mixture fit.....	97
Figure 4-19: New 2-Weibull mixture fit vs. 2-Weibull mixture fits of Castet & Saleh and Saleh & Castet....	97
Figure 4-20: Difference of the new 2-Weibull fit to Castet & Saleh and Saleh & Castet.....	98
Figure 4-21: PNZ-modified 2-Weibull mixture fit	99
Figure 4-22: Satellites failed in the first 14.5 years due to the PNZ-modified 2-Weibull mixture fit.	100
Figure 4-23: Weibull-plot and Box-plot of the PNZ-modified 2-Weibull mixture fit	100
Figure 4-24: Failures of Small, Medium and Large Satellites over 14.5 years.....	102
Figure 4-25: Failures of Small, Medium and Large Satellites over 5 years.....	102
Figure 4-26: Failures of Small, Medium and Large Satellites over the first year.	103
Figure 4-27: Nonparametric estimation of the complete set of Small, Medium and Large Satellites	103
Figure 4-28: Nonparametric estimation with two cut-offs	104
Figure 4-29: Small Satellites failed in the first 14.5 years due to the 2-Weibull function of Dubos et al. ...	105
Figure 4-30: Fraction failed due to the infant mortality term of Dubos et al	105
Figure 4-31: Fraction failed due to the wear-out term of Dubos et al.....	106
Figure 4-32: New PNZ modified Single-Weibull fit of Small Satellite reliability	106
Figure 4-33: SmallSats failed in the first 14.5 years due to the new PNZ modified Single-Weibull fit	107
Figure 4-34: Weibull-plot and Box-plot of the new PNZ modified Single-Weibull fit	107
Figure 4-35: PNZ modified 2-Weibull fit of Small Satellite reliability with cut-off at $t = 10.27$ years	108
Figure 4-36: PNZ modified Single-Weibull fit of SmallSat reliability with cut-off at $t = 10.27$ years	109
Figure 4-37: Small Satellites failed in the first 11 years due to the new PNZ Single-Weibull function.	109
Figure 4-38: Fraction failed due to the increasing failure rate term of the new Single-Weibull fit.....	110
Figure 4-39: Weibull-plot and Box-plot of the PNZ modified Single-Weibull fit	110
Figure 4-40: PNZ modified 2-Weibull mixture fit vs. PNZ modified Single-Weibull fit.	111
Figure 4-41: PNZ modified Single-Weibull fit vs. 2-Weibull mixture fit by Dubos et al.....	111
Figure 4-42: Difference of 2-Weibull fit of Dubos et al. to the new PNZ-modified Single-Weibull fit.....	112
Figure 4-43: Medium-sized satellites that failed within 14.5 years on-orbit due to the fit of Dubos et al. ..	113

Figure 4-44: Fraction failed due to the infant mortality term of Dubos et al	113
Figure 4-45: Fraction failed due to the wear-out term of Dubos et al.....	114
Figure 4-46: New 2-Weibull mixture fit of medium-sized satellites with cut-off at t = 8.4 years.....	114
Figure 4-47: Medium-sized satellites failed in the first 9 years due to the new 2-Weibull mixture fit.....	115
Figure 4-48: Fraction failed due to the infant mortality term of the new 2-Weibull mixture function	115
Figure 4-49: Fraction failed due to the wear-out term of the new 2-Weibull mixture function.....	116
Figure 4-50: Weibull-plot and Box-plot of the new 2-Weibull mixture fit	116
Figure 4-51: New 2-Weibull mixture fit vs. 2-Weibull mixture fit by Dubos et al.....	117
Figure 4-52: Difference of 2-Weibull mixture model of Dubos et al. to new 2-Weibull mixture fit.....	117
Figure 4-53: Large satellites that failed within 14.5 years on-orbit due to the fit of Dubos et al	118
Figure 4-54: Fraction that failed due to the infant mortality term of the 2-Weibull fit by Dubos et al	118
Figure 4-55: Fraction that failed due to the wear-out term of the 2-Weibull fit by Dubos et al.....	119
Figure 4-56: New PNZ modified 2-Weibull mixture fit of large satellite reliability.....	120
Figure 4-57: Large satellites failed in the first 12 years due to the new PNZ 2-Weibull mixture fit.....	120
Figure 4-58: Fraction failed due to the constant failure rate term of the new PNZ 2-Weibull mixture fit....	121
Figure 4-59: Fraction failed due to the wear-out term of the new PNZ 2-Weibull mixture fit	121
Figure 4-60: Weibull-plot and Box-plot of the new PNZ 2-Weibull mixture fit	122
Figure 4-61: PNZ-modified 2-Weibull mixture fit vs. 2-Weibull mixture fit by Dubos et al.....	122
Figure 4-62: Nonparametric estimation of CubeSat reliability and 95% confidence interval (1 year).....	124
Figure 4-63: Nonparametric reliability and PNZ-modified Single-Weibull fit of CubeSat reliability	125
Figure 4-64: CubeSats failed due to the PNZ-modified Single-Weibull function within 1.6 years	125
Figure 4-65: CubeSats failed due to the infant mortality term of the PNZ Single-Weibull function.....	126
Figure 4-66: Weibull-plot and Box-plot of the PNZ Single-Weibull fit of CubeSat reliability	126
Figure 4-67: Nonparametric reliability and PNZ-modified 2-Weibull mixture fit of CubeSat reliability.	127
Figure 4-68: CubeSats failed due to the PNZ-modified 2-Weibull mixture function.....	128
Figure 4-69: CubeSats that failed due to the infant mortality term of the PNZ 2-Weibull mixture fit.....	128
Figure 4-70: Weibull-plot and Box-plot of the PNZ-modified 2-Weibull mixture fit.....	129
Figure 4-71: CubeSats failed due to the PNZ-modified 2-Weibull mixture fit (1 year)	129
Figure 4-72: Comparison of two PNZ-modified 2-Weibull mixture fits	130
Figure 4-73: MLE vs. and nonlinear least-squares fitted 2-Weibull mixture functions	131
Figure 4-74: CubeSats failed due to MLE estimated PNZ-modified 2-Weibull mixture fit	131
Figure 4-75: Subsystem contributions to CubeSat failure after ejection (DOA, 30 days and 90 days)	132
Figure 4-76: Nonparametric and Parametric Modelling “unknown” section and EPS for CubeSats.....	132
Figure 4-77: Nonparametric and Parametric Modelling OBC and COM for CubeSats.....	133
Figure 4-78: Developers’ beliefs likelihood of failure for own mission general university-built CubeSat ...	133
Figure 4-79: Knowledge level on and implementation of risk & failure analysis on CubeSats	134
Figure 4-80: First-MOVE in launch & deployed configuration and its sensor position	137
Figure 4-81: MOVE-II in launch configuration and deployed configuration.....	139
Figure 4-82: Explosion Drawing of MOVE-II	140
Figure 4-83: Measurement circuit of the PL on MOVE-II & Top side of the CubeSat.....	140
Figure 4-84: Toppanel, Mainpanel and Sidepanels of the ADCS on the MOVE-II CubeSat	141
Figure 4-85: UHF/VHF Transceiver and S-Band Transceiver of MOVE-II.....	142
Figure 4-86: Overview of the Software and Interfaces of MOVE-II.....	143
Figure 4-87: Structure of MOVE-II.....	143
Figure 4-88: Overview of 2SMARD and working principle of 2SMARD.....	144
Figure 4-89: Thermal Prototype of MOVE-II and TVAC of LRT.....	145
Figure 4-90: Thermal Prototype of MOVE-II and Thermal-Vacuum Chamber of LRT	145
Figure 4-91: Subsystem-level test in TVAC and System Level testing of the MOVE-II FM.....	146
Figure 4-92: Calculated temperature distribution of the MOVE-II ESATAN model.....	147
Figure 4-93: 3D Printed Prototype of MOVE-II.....	148
Figure 4-94: Missions operations interface of the MOVE-II mission	150

Figure 4-95: The PC/104 compatible carrier board for a Beaglebone Black Wireless.....	151
Figure 4-96: ADCS Control Loop.....	151
Figure 4-97: Mainpanel in the HiL-Testbed between the ADCS Testing Board and the Panel Emulator ..	152
Figure 4-98: B-Dot Detumbling Controller Spinning Down.....	152
Figure 4-99: Sun-Pointing Controller adjusting spin and reducing the pointing error	153
Figure 4-100: Remote testing setup of MOVE-II	154
Figure 4-101: Rendering and photo of MOVE-II FlatSat	155
Figure 4-102: Example of a bug-ticket of the MOVE-II bug tracking system	157
Figure 4-103: Workflow of the automated bug tracking in MOVE-II.....	158
Figure 4-104: Typical appearance of the automated bug handling application “Elfriede” in Slack	158
Figure 4-105: Cumulative number of errors on the FM and EM (the space segment) of MOVE-II.....	160
Figure 4-106: Analysis of specific regions of the number of cumulative errors.....	161
Figure 4-107: Notification Service of the MOVE-II Slack interface.....	161
Figure 4-108: Origin of all bugs detected in the space segment of MOVE-II.....	162
Figure 4-109: Software errors detected on the space segment of MOVE-II over time.	162
Figure 4-110: Hardware errors detected on the space segment of MOVE-II over time.....	163
Figure 4-111: All errors detected in the MOVE-II system.....	163
Figure 4-112: Origin of all bugs detected in the MOVE-II project.....	164
Figure 4-113: Cumulative errors of the mission operations interface and the ground station.....	164
Figure 4-114: Cumulative errors of the CDH subsystem and the ADCS subsystem	165
Figure 4-115: Cumulative errors of the COM subsystem and the EPS subsystem	165
Figure 4-116: Cumulative errors of the GPS subsystem and the PL subsystem	166
Figure 4-117: Cumulative errors of the STR subsystem and the THM subsystem.....	166
Figure 4-118: Cumulative errors of the GSE of MOVE-II.	167
Figure 4-119: Fraction of errors occurring on different systems of the MOVE-II project.....	167
Figure 4-120: Cumulative number of critical failures of MOVE-II over testing time	168
Figure 4-121: Origin of critical failures occurring in the MOVE-II project	168
Figure 4-122: Fit of a basic exponential reliability growth model to errors of the space segment.	170
Figure 4-123: Stability of the prediction of the basic exponential reliability growth model	170
Figure 4-124: Prediction of basic exponential model with different time ranges.	171
Figure 4-125: Fit of the Yamada & Osaki reliability growth model	172
Figure 4-126: Stability of the prediction of the Yamada & Osaki reliability growth model.....	172
Figure 4-127: Fit of the exponential reliability growth model.....	173
Figure 4-128: Stability of the prediction of the exponential reliability growth model	174
Figure 4-129: Prediction of the exponential reliability growth model with variable starting date	174
Figure 4-130: Number of estimated errors over time delayed S-shaped reliability growth model	175
Figure 4-131: Fit of the delayed S-shaped software reliability growth model	175
Figure 4-132: Stability of the prediction of the delayed S-shaped software reliability growth model	176
Figure 4-133: Prediction of the delayed S-shaped software reliability growth model	176
Figure 4-134: Fit of the inflection S-shaped software reliability growth model.	178
Figure 4-135: Stability of the prediction of the inflection S-shaped software reliability growth model.	178
Figure 4-136: Prediction of the inflection S-shaped software reliability growth model.....	179
Figure 4-137: Comparison of the estimation of all models	179
Figure 4-138: Fit of the basic exponential reliability growth model to the critical failures of MOVE-II.....	180
Figure 4-139: Stability of the prediction of critical failures of the space segment of MOVE-II	181
Figure 4-140: Prediction of critical failures by the basic exponential reliability growth model.....	181
Figure 4-141: Fit of the exponential model with variable starting date to the critical failures.....	182
Figure 4-142: Stability of critical failure prediction of the exponential model with variable starting date. ..	182
Figure 4-143: Prediction of critical failures by the exponential model with variable starting date	183
Figure 4-144: Comparison of the estimation of all models	183
Figure 4-145: Reliability estimation of MOVE-II based on the number of critical failures	185

Figure 4-146: Estimated reliability if the system level tests would continue after day 319	186
Figure 4-147: User Interface of the tool FREE MTBF calculator.....	187
Figure 4-148: MOVE-II Sidepanel front- and backside.	188
Figure 5-1: Number of man-made objects in space over time	199
Figure 5-2: Working hours voluntarily spent & logged by students of the MOVE-II team.....	201
Figure 6-1: Failure Rate of the modified Weibull fit by Peng & Zhang	253
Figure 6-2: PNZ-modified 2-Weibull mixture fit vs. fits of Castet & Saleh and Saleh & Castet.....	253
Figure 6-3: Difference 2-Weibull fits of Castet & Saleh and Saleh & Castet to PNZ 2-Weibull fit	254
Figure 6-4: PNZ-modified Single-Weibull mixture fit vs. 2-Weibull mixture models of Dubos et al.....	254
Figure 6-5: Difference of 2-Weibull fit of Dubos et al. to the PNZ-modified Single-Weibull fit	255
Figure 6-6: SmallSats that failed due to the constant failure rate term of the PNZ Single-Weibull fit.....	255
Figure 6-7: Fit of the Yamada & Osaki model to the critical failures of the space segment of MOVE-II....	256
Figure 6-8: Stability of the Yamada & Osaki prediction of critical failures of MOVE-II.....	256
Figure 6-9: Prediction of critical failures by the Yamada & Osaki model	257
Figure 6-10: Fit of the Delayed S-shaped reliability growth model to the critical failures of MOVE-II.....	257
Figure 6-11: Stability of the Delayed S-shaped reliability growth prediction of critical failures	258
Figure 6-12: Prediction of critical failures by the Delayed S-shaped reliability growth model.....	258
Figure 6-13: Fit of the Inflection S-shaped reliability growth model to the critical failures of MOVE-II	258
Figure 6-14: Stability of Inflection S-shaped reliability growth model prediction of critical failures	259
Figure 6-15: Prediction of critical failures by the Inflection S-shaped reliability growth model	260
Figure 6-16: Comparison of the estimation of all models.....	260

A.2 List of Tables

Table 2-1: Measures of reliability	12
Table 2-2: Statistical distributions used in reliability engineering.....	14
Table 2-3: Programmatic concerns of the effects of the space environment on spacecraft	22
Table 2-4: Space environment effects on spacecraft subsystems.	23
Table 2-5: Overview of handbook based reliability prediction methods	43
Table 4-1: The CFDB	123
Table 4-2: Failure times (in days) of all CubeSats launched up to 06/30/2014.	123
Table 4-3: Reliability Prediction for the MOVE-II Sidepanel.....	189
Table 5-1: Total number of failures predicted by the five different models	203
Table 5-2: Total number of critical failures predicted by the five different models	204
Table 5-3: Total number of critical failures predicted by the five different models	205
Table 6-1: Distribution functions used in reliability analysis	246
Table 6-2: Comparison of handbook based reliability prediction methods	247
Table 6-3: Analysis of reliability prediction methods used in traditional spaceflight.....	248
Table 6-4: Recommendations for handbook based reliability prediction methods	249
Table 6-5: Typical structures and their reliability function	250
Table 6-6: Comparison of test conditions for military, spaceflight and automotive parts	251
Table 6-7: Nonparametric reliability estimation by Castet & Saleh	252
Table 6-8: Main techniques to promote teamwork in small satellite and CubeSat projects.....	261

Appendix B Supplementary Figures and Tables

Table 6-1: Distribution functions used in reliability analysis, adapted from [39]

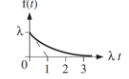
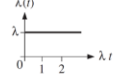
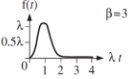
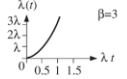
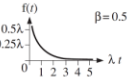
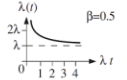
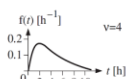
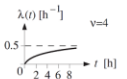
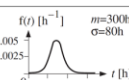
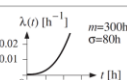
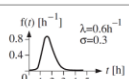
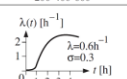
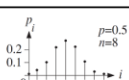

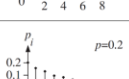
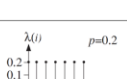
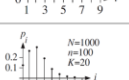
Name	Distribution Function $F(t) = \Pr\{\tau \leq t\}$	Density $f(t) = dF(t)/dt$	Parameter Range	Failure Rate $\lambda(t) = f(t) / (1 - F(t))$	Mean $E[\tau]$	Variance $\text{Var}[\tau]$	Properties
Exponential	$1 - e^{-\lambda t}$		$t > 0$ ($F(t)=0, t \leq 0$) $\lambda > 0$		$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$	Memoryless: $\Pr\{\tau > t + x_0 \mid \tau > x_0\} = \Pr\{\tau > t\} = e^{-\lambda t}$
Weibull	$1 - e^{-(\lambda t)^\beta}$		$t > 0$ ($F(t)=0, t \leq 0$) $\lambda, \beta > 0$		$\frac{\Gamma(1 + \frac{1}{\beta})}{\lambda}$	$\frac{\Gamma(1 + \frac{2}{\beta}) - \Gamma^2(1 + \frac{1}{\beta})}{\lambda^2}$	Monotonic failure rate, strictly increasing for $\beta > 1$ ($\lambda(+0)=0, \lambda(+\infty)=\infty$), decreasing for $\beta < 1$ ($\lambda(+0)=\infty, \lambda(+\infty)=0$)
Gamma	$\frac{1}{\Gamma(\beta)} \int_0^t x^{\beta-1} e^{-x} dx$		$t > 0$ ($F(t)=0, t \leq 0$) $\lambda, \beta > 0$		$\frac{\beta}{\lambda}$	$\frac{\beta}{\lambda^2}$	Laplace transf. exists: $\tilde{f}(s) = \lambda^\beta / (s + \lambda)^\beta$; Monotonic failure rate with $\lambda(+\infty) = \lambda$; Exp. for $\beta=1$, Erlangian for $\beta=n=2, 3, \dots$ (sum of n exp. distrib. random variables $1/\lambda$)
Chi-square (χ^2)	$\frac{\int_0^t x^{v/2-1} e^{-x/2} dx}{2^{v/2} \Gamma(v/2)}$		$t > 0$ ($F(t)=0, t \leq 0$) $v = 1, 2, \dots$ (degrees of freedom)		v	$2v$	Gamma with $\beta = v/2, v=1, 2, \dots$ and $\lambda = 1/2$; for $v=2, 4, \dots \Rightarrow F(t) = 1 - \sum_{i=0}^{v/2-1} \frac{(t/2)^i}{i!} e^{-t/2}$ (sum of $v/2$ exp. distrib. random var. $1/\lambda=1/2$)
Normal	$\frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^t e^{-(x-m)^2/2\sigma^2} dx$		$-\infty < t, m < \infty$ $\sigma > 0$		m	σ^2	$F(t) = \Phi((t-m)/\sigma)$ $\Phi(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^t e^{-x^2/2} dx$
Lognormal	$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\ln(t)} e^{-x^2/2} dx$		$t > 0$ ($F(t)=0, t \leq 0$) $\lambda, \sigma > 0$		$\frac{e^{\sigma^2/2}}{\lambda}$	$\frac{e^{2\sigma^2} - e^{\sigma^2}}{\lambda^2}$	$F(t) = \Phi(\ln(\lambda t)/\sigma)$; In τ has a normal distribution with $m = \ln(1/\lambda) = E[\ln \tau]$ and $\sigma^2 = \text{Var}[\ln \tau]$
Binomial	$\Pr\{\zeta \leq k\} = \sum_{i=0}^k p_i$ $p_i = \binom{n}{i} p^i (1-p)^{n-i}$		$k = 0, \dots, n$ $0 < p < 1$	not relevant	np	$np(1-p)$	$p_i = \Pr\{i \text{ successes in } n \text{ Bernoulli trials}\}$ (n independent trials with $\Pr\{A\} = p$); Random sample with replacement
Poisson	$\Pr\{\zeta \leq k\} = \sum_{i=0}^k p_i$ $p_i = \frac{m^i}{i!} e^{-m}$		$k = 0, 1, \dots$ $m > 0$	not relevant	m	m	$m = \lambda t \Rightarrow (\lambda t)^i e^{-\lambda t} / i! = \Pr\{i \text{ failures in } (0, t] \mid \lambda\}$; $\binom{n}{i} p^i (1-p)^{n-i} = \frac{(np)^i}{i!} e^{-np}$
Geometric	$\Pr\{\zeta \leq k\} = \sum_{i=1}^k p_i = 1 - (1-p)^k$ $p_i = p(1-p)^{i-1}$		$k = 1, 2, \dots$ $0 < p < 1$		$\frac{1}{p}$	$\frac{1-p}{p^2}$	Memoryless: $\Pr\{\zeta > i + j \mid \zeta > i\} = (1-p)^j$; $p_i = \Pr\{\text{first success in a sequence of Bernoulli trials occurs at the } i\text{th trial}\}$
Hyper-geometric	$\Pr\{\zeta \leq k\} = \sum_{i=0}^k \frac{\binom{K}{i} \binom{N-K}{n-i}}{\binom{N}{n}}$		$k = 0, 1, \dots, \min(K, n)$	not relevant	$n \frac{K}{N}$	$\frac{K n (N - K) (N - n)}{N^2 (N - 1)}$	Random sample without replacement

Table 6-2: Comparison of handbook based reliability prediction methods. Source: [138]

Questions for Comparison	MIL-HDBK-217F	Telcordia SR-332	217Plus	PRISM	FIDES
Does the methodology identify the sources used to develop the prediction methodology and describe the extent to which the source is known?	Yes	Yes	Yes	Yes	Yes
Are assumptions used to conduct the prediction according to the methodology identified, including those used for the unknown data?	Yes	Yes	Yes	Yes	Yes (must pay for modeling software).
Are sources of uncertainty in the prediction results identified?	No	Yes	Yes	No	Yes
Are limitations of the prediction results identified?	Yes	Yes	Yes	Yes	Yes
Are failure modes identified?	No	No	No	No	Yes, the failure mode profile varies with the life profile.
Are failure mechanisms identified?	No	No	No	No	Yes
Are confidence levels for the prediction results identified?	No	Yes	Yes	No	No
Does the methodology account for life-cycle environmental conditions, including those encountered during (a) product usage (including power and voltage conditions), (b) packaging, (c) handling, (d) storage, (e) transportation, and (f) maintenance conditions?	No. It does not consider the different aspects of environment. There is a temperature factor π_T and an environment factor π_E in the prediction equation.	Yes, for normal use life of the product from early life to steady-state operation over the normal product life.	No	No	Yes. It considers all of the life-cycle environmental conditions.
Does the methodology account for materials, geometry, and architectures that comprise the parts?	No	No	No	No	Yes, when relevant materials, geometry and such are considered in each part model.
Does the methodology account for part quality?	Quality levels are derived from specific part-dependent data and the number of the manufacturer screens the part goes through.	Four quality levels that are based on generalities regarding the origin and screening of parts.	Quality is accounted for in the part quality process grading factor.	Part quality level is implicitly addressed by process grading factors and the growth factor, p_G .	Yes
Does the methodology allow incorporation of reliability data and experience?	No	Yes, through Bayesian method of weighted averaging.	Yes, through Bayesian method of weighed averaging	Yes, through Bayesian method of weighted averaging.	Yes. This can be done independently of the prediction methodology used.
Input data required for the analysis	Information on part count and operational conditions (e.g., temperature, voltage; specifics depend on the handbook used).				
Other requirements for performing the analysis	Effort required is relatively small for using the handbook method and is limited to obtaining the handbook.				
What is the coverage of electronic parts?	Extensive	Extensive	Extensive	Extensive	Extensive
What failure probability distributions are supported?	Exponential	Exponential	Exponential	Exponential	Phase Contributions

Table 6-3: Analysis of reliability prediction methods used in traditional spaceflight. Source: [154]

Feature	Methodology						
	FTA	FMEA /FMECA	RBD	Parts Count Prediction	Parts Stress Prediction	Monte Carlo Simulation	Markov
Strengths:							
Top down approach, so good at assessing higher level issues	✓	✗	✗	(✓) – mainly used for initial assessment and for apportionment	✓	(✓) – not specifically top-down, but could be used	✗
Bottom up approach, so good at assessing part/ components	✗	✓	✗	✓	(✓) – used when part stress analysis has been performed, usually prior to CDR	(✓) – not specifically bottom-up, but could be used	✗
Can assess combination of events	✓	✗	(✓) – only in terms of active and backup chains	(✓) – only in terms of active and backup chains	✓	✓	(✓) – only in terms of changing between states
Can be used as a qualitative or quantitative method	✓	(✓) – failure rates not always used (FMEA)	✓	(✓) – only failure rates used, not qualitative	✓	(✓) – only failure rates used, not qualitative	(✓) – only failure rates used, not qualitative
Identifies weak points in the design / supports design improvement	✓	✓	✓	✓	✓	✓	✓
Includes common mode failures	✓	✓	✗	✗	✗	(✓) – may be difficult to implement	✗
Simple, graphical output	✓	✗	✓	✗	✗	✗	✓
Widely used in other domains (FTA - nuclear, chemical plants, FMEA – military, avionics)	✓	✓	(✓) – probably not used in all domains		✓	✓	(✓) – probably not used in all domains
Widely used in other domains (FTA - nuclear, chemical plants, FMEA – military, avionics)	✓	✓	(✓) – probably not used in all domains	✓	✓	✓	(✓) – probably not used in all domains
Able to Identify critical fault paths, and sequences of events	✓	(✓) – event sequences not covered	(✓) – shows success paths, not sequences	✗	✗	(✓) – may be difficult to implement	(✓) – may be difficult to implement
Good at identifying failure propagation	✗	✓	✗	✗	✗	✗	✗
Good at identifying single points of failure	✗	✓	✓	✗	✗	✗	✗
HW and SW failures able to be assessed	✓	✓	(✓) – not normally used for SW	(✓) – not normally used for SW	✗	✓	✓
Weaknesses:							
Generally, detailed knowledge of the system is required	✓	✓	✗	✗	✗	✗	✗
Potential difficulty in interpretation at different levels	✓	✓	✗	✗	✗	✗	✗
Time consuming	✓	✗	✗	✓	✓	✓	✓
Very time consuming	✗	✓	✗	✗	dependent on the number of components to be assessed	✗	✗
Results not always believed	(✓) – some failure modes may not be considered realistic	(✓) – some failure modes may not be considered realistic	✗	✓		✓	✓

Table 6-4: Recommendations for handbook based reliability prediction methods by Airbus Defence and Space. Source: [155]

Parts	MIL-HDBK	FIDES	217+	RECOMMENDATION
Capacitors	1	2	3	MIL-HDBK
Connectors	3	1	2	FIDES
Piezoelectric	2	1	N/A	FIDES
Diodes	2	1	3	FIDES
Filters	To be considered at part level – else FIDES			
Fuses	N/A	1	N/A	FIDES
Inductors	2	1	3	FIDES
ICs	2	1	3	FIDES
Relays	1	2	3	MIL-HDBK
Resistors	2	1	3	FIDES
Thermistors	1	N/A	2	MIL-HDBK
Transistors	2	1	3	FIDES
Transformers	2	1	3	FIDES
Switches	2	1	N/A	FIDES
Optoelectronics	2	1	3	FIDES
Thermostats	N/A	N/A	N/A	In house
Hybrids	2	1	X	FIDES
Battery cells	N/A	N/A	N/A	In house
Solar cells	N/A	N/A	N/A	In house

Table 6-5: Typical structures and their reliability function. Source: [39]

Reliability Block Diagram	Reliability Function ($R_S = R_{S0}(t)$; $R_i = R_i(t)$, $R_i(0)=1$)	Remarks
1 	$R_S = R_i$	One -item structure, $\lambda(t) = \lambda \Rightarrow R_i(t) = e^{-\lambda t}$
2 	$R_S = \prod_{i=1}^n R_i$	Series structure, $\lambda_S(t) = \lambda_1(t) + \dots + \lambda_n(t)$
3 	$R_S = R_1 + R_2 - R_1 R_2$	1-out-of-2-redundancy, $R_1(t) = R_2(t) = e^{-\lambda t}$ $\Rightarrow R_S(t) = 2e^{-\lambda t} - e^{-2\lambda t}$
4 	$E_1 = \dots = E_n = E$ $\rightarrow R_1 = \dots = R_n = R$ $R_S = \sum_{i=k}^n \binom{n}{i} R^i (1-R)^{n-i}$	k-out-of-n redundancy for $k = 1$ $\Rightarrow R_S = 1 - (1-R)^n$ see p. 44 for $E_1 \neq \dots \neq E_n$
5 	$R_S = (R_1 R_2 R_3 + R_4 R_5 - R_1 R_2 R_3 R_4 R_5) R_6 R_7$	Series - parallel structure
6 	$E_1 = E_2 = E_3 = E$ $\rightarrow R_1 = R_2 = R_3 = R$ $R_S = (3R^2 - 2R^3) R_v$	Majority redundancy, general case (n+1)-out-of-(2n+1), n=1,2,...
7 	$R_S = R_5 (R_1 + R_2 - R_1 R_2) \cdot (R_3 + R_4 - R_3 R_4) + (1 - R_5) \cdot (R_1 R_3 + R_2 R_4 - R_1 R_2 R_3 R_4)$	Bridge structure (bi-directional on E_5)
8 	$R_S = R_4 [R_2 + R_1 (R_3 + R_5 - R_3 R_5) - R_1 R_2 (R_3 + R_5 - R_3 R_5)] + (1 - R_4) R_1 R_3$	Bridge structure (unidirectional on E_5)
9 	$R_S = R_2 R_1 (R_4 + R_5 - R_4 R_5) + (1 - R_2) R_1 R_3 R_5$	The element E_2 appears twice in the reliability block diagram (not in the hardware)

Table 6-6: Comparison of test conditions for military, spaceflight and automotive parts. Source: [26]

Test condition	MIL-STD-883 requirement IAW ³ MIL-PRF-38535	NASA GSFC 311-INST requirement	Automotive requirement
Visual inspection / physical size compliance	Inspected at 7-10X magnification for no visible defects	Inspected at 7-10X magnification for no visible defects	Inspected for no visible defects
Soldering heat resistance	Steam age 8 hours; bake 100°C 1 hour; solder test by dip in solder pot 245°C ±5C, 5 seconds	Steam age 8 hours; bake 100°C 1 hour; solder test by dip in solder pot 245°C ±5C, 5 seconds	Moisture 30°C 70%RH 168 hours; bake 125°C 24 hours; solder test by infrared reflow 235°C±5C, 10 seconds
Thermal shock	-55° to 125°C, 15 cycles, 10 second transfer time, 2 minute dwell	Not required	-65° to 150°C, 500 cycles, 10 minutes/cycle
Particle Impact Noise Detection	Not required	3 1,000g shocks, 3 second 20g 40-250Hz vibration, repeated 4 times	Not required
High temperature baking	IAW device specification	IAW device specification	150°C, 2000 hours
Span of life	125°C, 1000 hours, all Vcc's at max rated voltage	Max junction temperature, 1000 hours, all Vcc's at max rated voltage, performing operational test	150°C, 2000hours, Vcc5V=6.5V, Vcc3.3V=4.2V, performing operational test
Lead integrity	0.229kg, 90° bend, 3 times	0.229kg, 90° bend, 3 times	Not required
Salt atmosphere (corrosion)	35°C, 95%RH, 24 hours, 20,000 to 50,000 mg/m ² NaCl deposited	Not required	Required but details unavailable
Temperature / humidity	130°C, 85%RH, 100 hours	Not required	85°C, 85%RH, 2000 hours, Vcc5V=5.5V, Vcc3.3V=3.6V, performing operational test
Solvent resistance	1 minute in: alcohol solvent at 25°C; organic solvent at suitable temperature; ethyl inorganic solvent at 70°C	1 minute in: alcohol solvent at 25°C; organic solvent at suitable temperature; ethyl inorganic solvent at 70°C	Not required
Autoclave (PCT)	121°C, 100%RH, 2atm, 96 hours, unbiased ⁴	Not required	130°C, 85%RH, 240 hours, Vcc5V=5.5V, Vcc3.3V=3.6V
Thermal cycling	-65° to 150°C, 100 cycles, 10 minute dwell	-65° to 150°C, 10 cycles, 10 minute dwell	-65° to 150°C, 1000 cycles, 1 hour/cycle
Static electric breakdown (Human Body Model)	C=100pF, R=1.5k, 3 samples; voltage breakdown range recorded	Not required	C=100pF, R=1.5k, 5 samples; voltage breakdown levels recorded
Static electric breakdown (Machine Model)	Not required	Not required	C=200pF, R=200, 5 samples; voltage breakdown levels recorded

Table 6-7: Nonparametric reliability estimation of the studied group of 1,584 satellites by Castet & Saleh.
 Source: [115]

Failure time t_i (year)	$\hat{R}(t_i)$	Failure time t_i (year)	$\hat{R}(t_i)$	Failure time t_i (year)	$\hat{R}(t_i)$
0.0027	0.9975	0.6899	0.9740	5.3854	0.9486
0.0055	0.9968	0.7420	0.9732	5.5003	0.9475
0.0082	0.9956	0.8460	0.9725	5.7248	0.9464
0.0110	0.9943	0.8597	0.9718	5.7413	0.9453
0.0137	0.9930	0.8679	0.9711	5.7440	0.9442
0.0192	0.9924	0.9144	0.9703	5.9713	0.9430
0.0246	0.9918	0.9966	0.9696	5.9986	0.9419
0.0329	0.9911	1.2731	0.9688	6.1246	0.9408
0.0411	0.9898	1.4100	0.9681	6.6502	0.9396
0.0438	0.9885	1.9055	0.9673	6.6639	0.9384
0.0630	0.9879	1.9192	0.9665	6.7680	0.9372
0.0986	0.9872	1.9521	0.9657	7.0554	0.9359
0.1396	0.9865	1.9767	0.9649	7.0637	0.9347
0.1451	0.9859	1.9822	0.9641	7.1841	0.9334
0.1752	0.9852	2.1547	0.9633	7.3977	0.9322
0.1862	0.9845	2.8830	0.9624	7.9863	0.9308
0.1999	0.9838	2.9377	0.9616	8.0684	0.9295
0.2163	0.9831	3.0719	0.9607	8.1123	0.9281
0.2437	0.9824	3.1376	0.9598	8.3176	0.9267
0.2793	0.9817	3.1951	0.9590	8.4244	0.9253
0.2930	0.9810	3.2416	0.9581	8.6489	0.9238
0.3368	0.9803	3.3758	0.9572	8.9473	0.9223
0.3504	0.9796	3.4387	0.9564	9.4593	0.9207
0.3587	0.9789	3.6879	0.9555	10.0862	0.9188
0.4572	0.9782	3.9918	0.9545	10.2916	0.9169
0.5202	0.9775	4.2464	0.9536	11.4771	0.9142
0.5394	0.9768	4.4819	0.9527	11.8385	0.9113
0.6051	0.9761	4.8679	0.9517	13.4401	0.9074
0.6270	0.9754	4.9199	0.9507	13.8070	0.9031
0.6489	0.9747	5.0267	0.9496	14.2560	0.8983

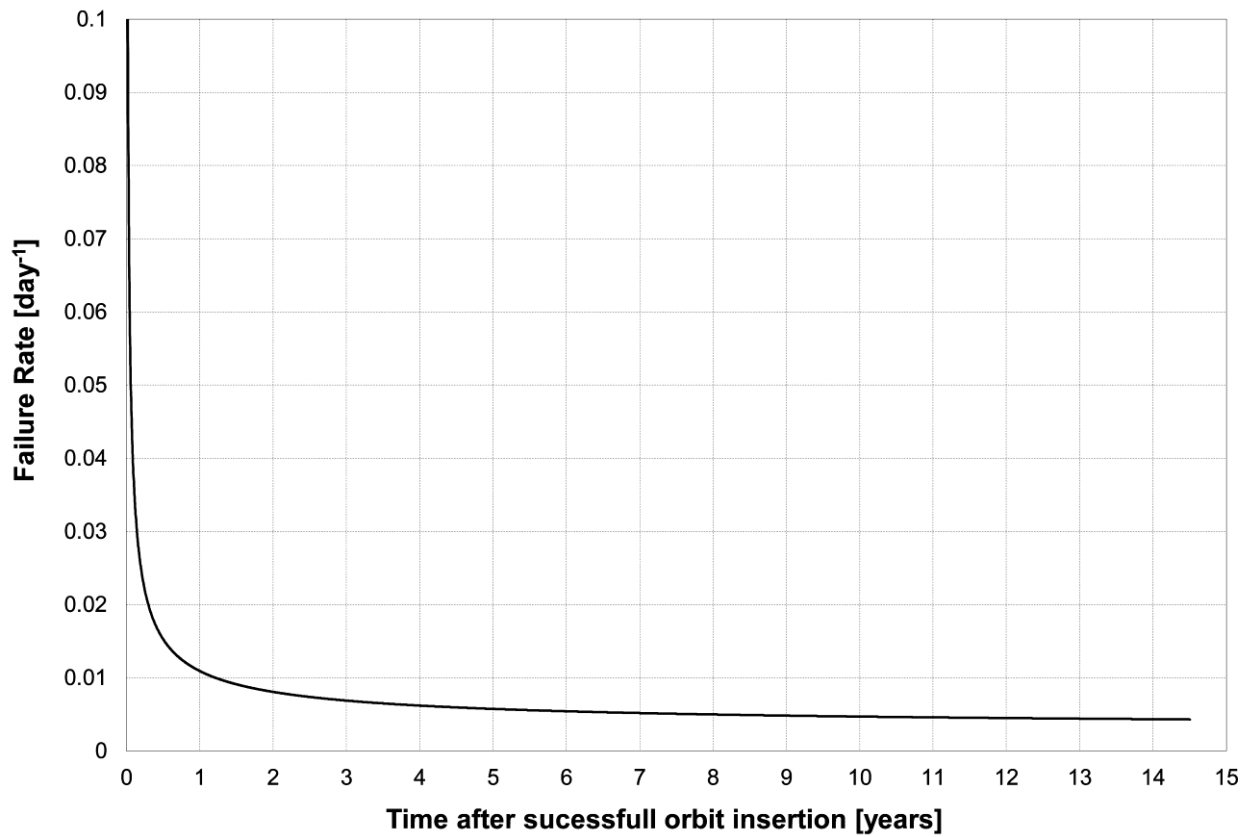


Figure 6-1: Failure Rate of the modified Weibull fit by Peng & Zhang [124].

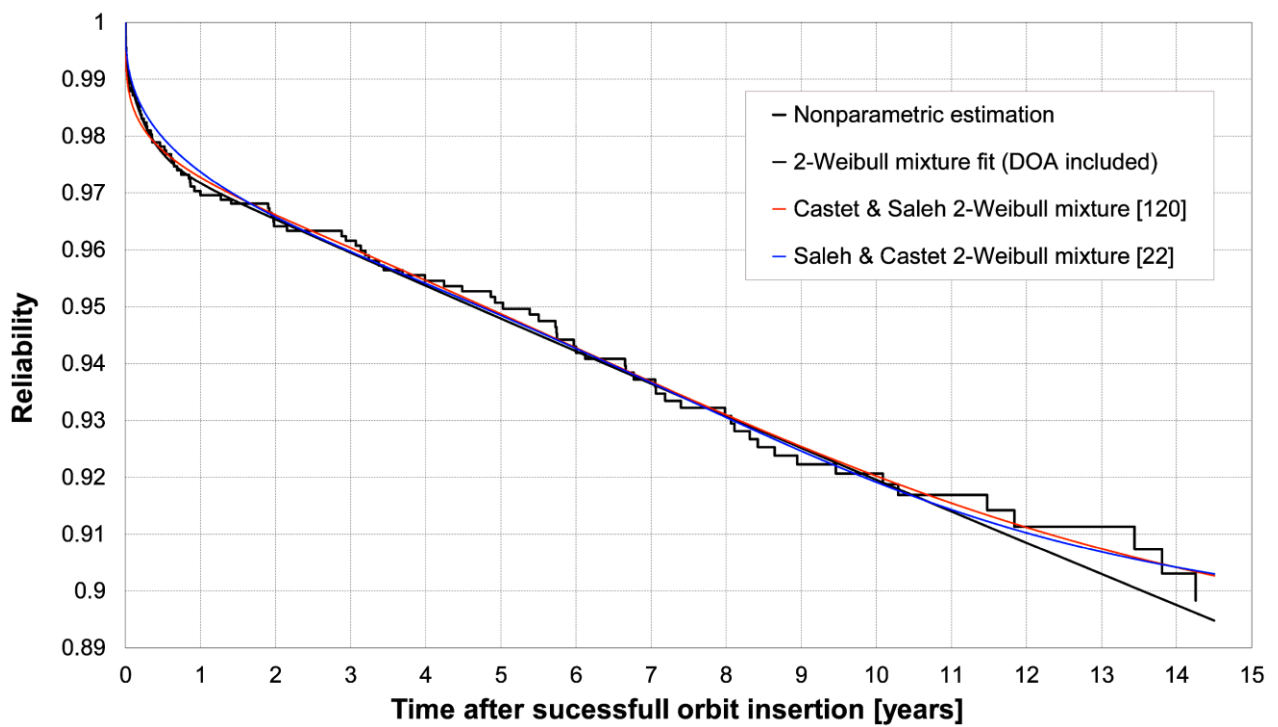


Figure 6-2: PNZ-modified 2-Weibull mixture fit (equation (33)) vs. 2-Weibull mixture models of Castet & Saleh [120] (equation (27)) and Saleh & Castet [22] (equation (28)). Data source of nonparametric estimation: [115].

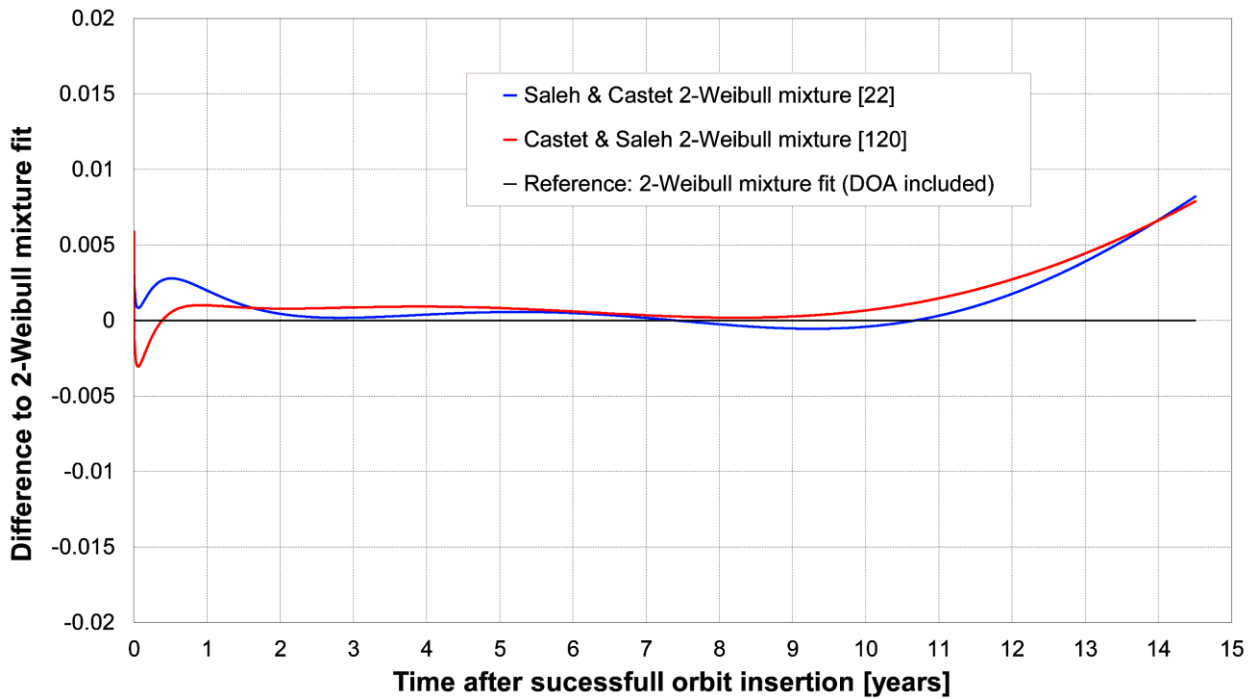


Figure 6-3: Difference of the 2-Weibull mixture models of Castet & Saleh [120] (equation (27)) and Saleh & Castet [22] (equation (28)) to the PNZ-modified 2-Weibull mixture fit (equation (33)). The deviations at $t = 0$ stem from the rate of satellites that are estimated to have never been in an operable state ($1 - p_{NZ} = 0.59\%$).

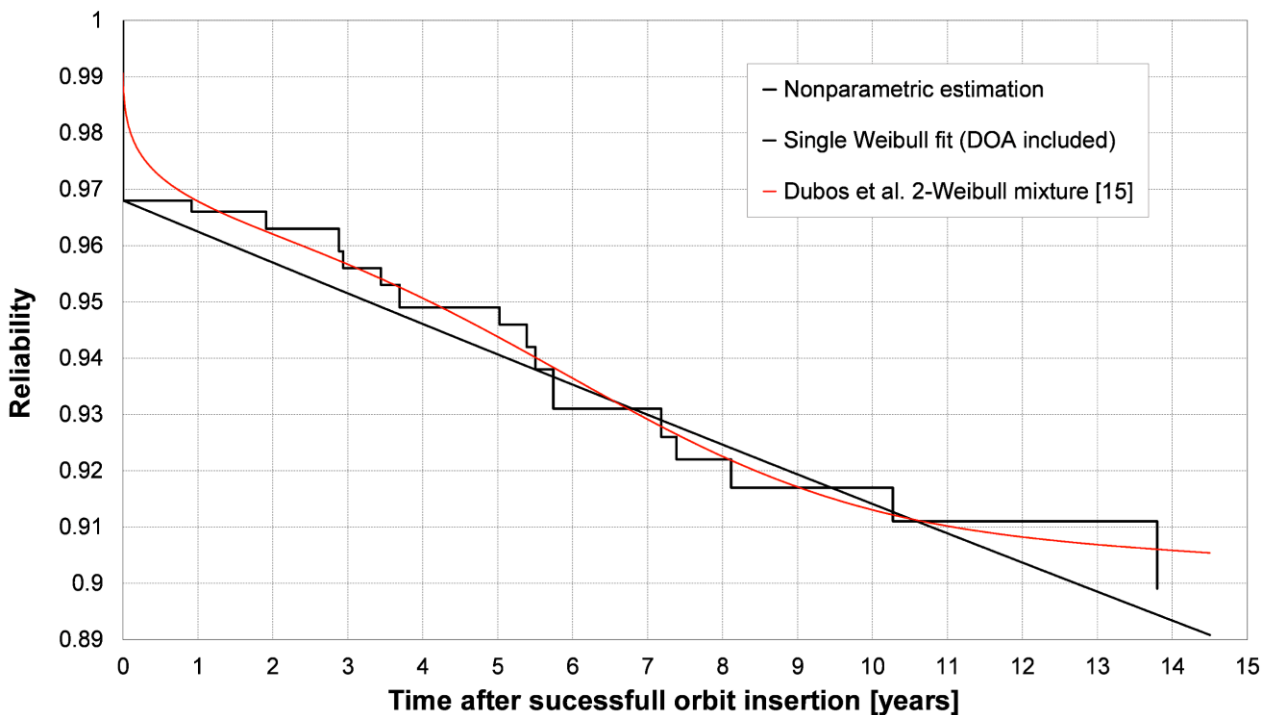


Figure 6-4: PNZ-modified Single-Weibull mixture (equation (35)) fit vs. 2-Weibull mixture models of Dubos et al. [15] (equation (34)). Data source of nonparametric estimation: [15]

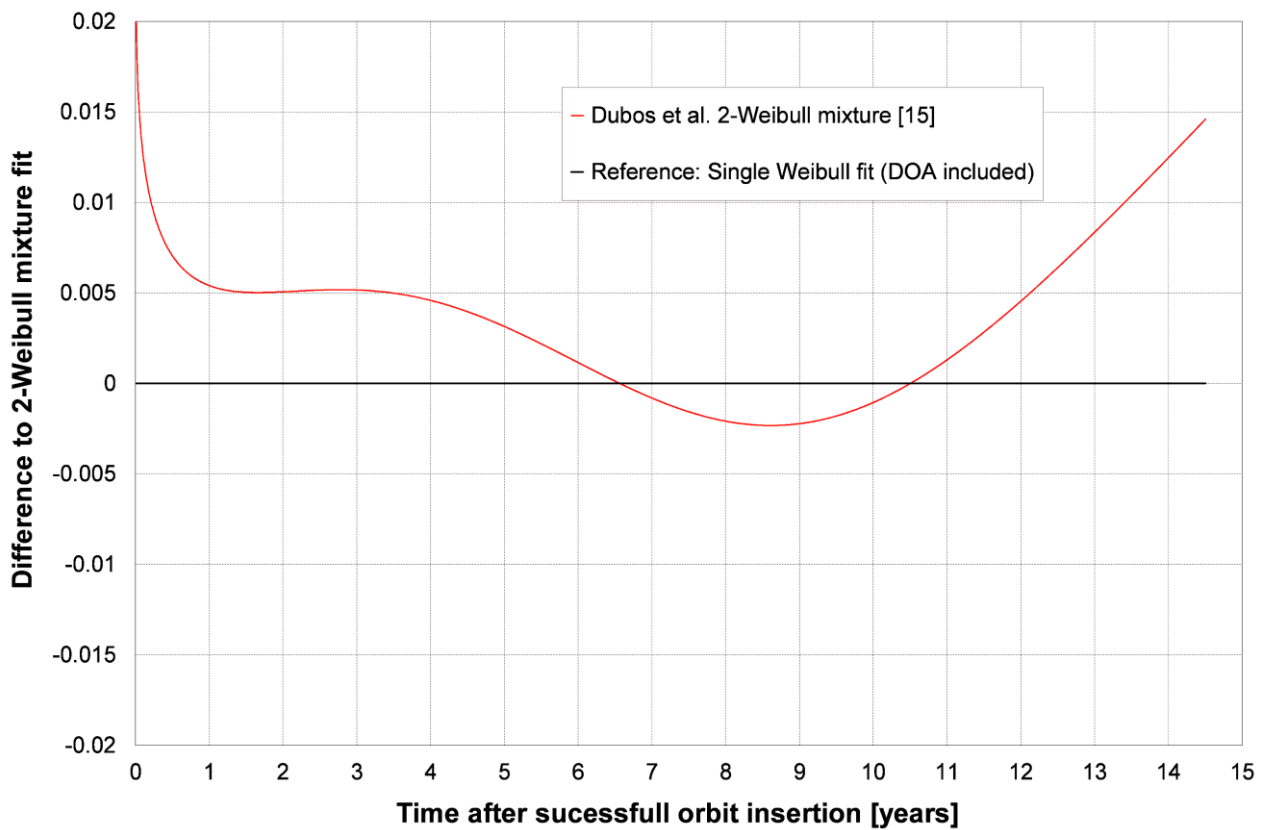


Figure 6-5: Difference of the 2-Weibull mixture model of Dubos et al. [15] (equation (34)) to the PNZ-modified Single-Weibull fit (equation (35)). The deviations at $t=0$ stem from the rate of satellites that are estimated to have never been in an operable state ($1 - p_{NZ} = 0.32\%$).

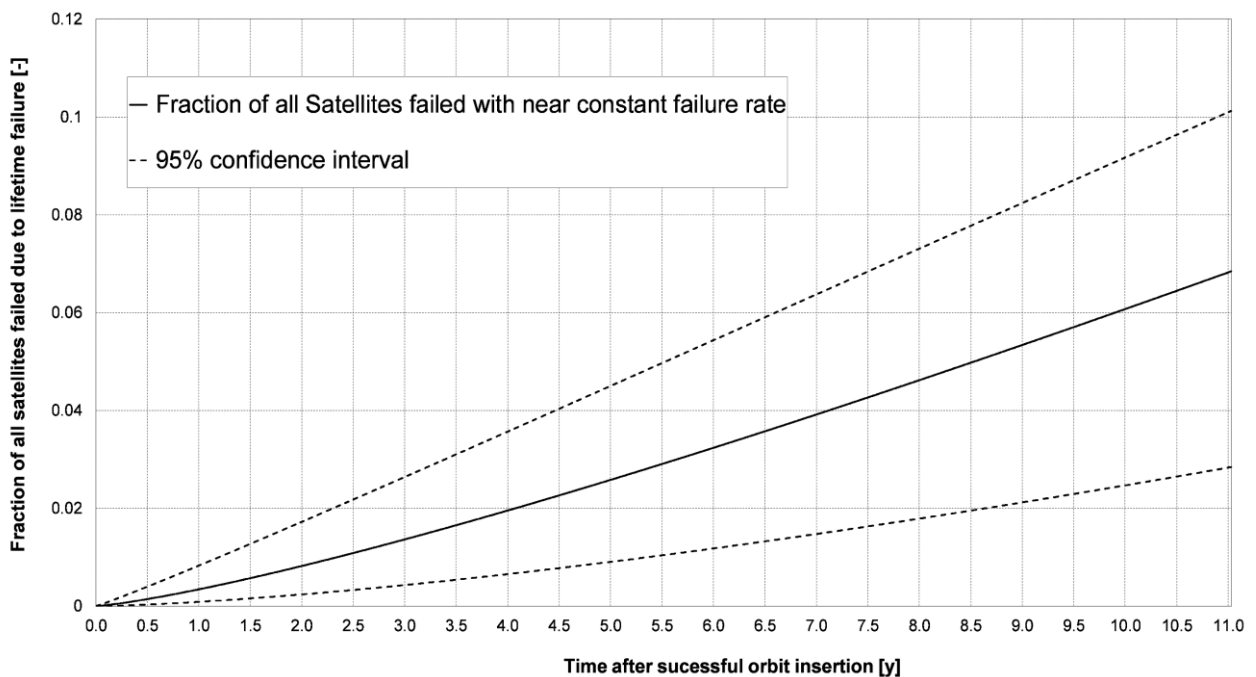


Figure 6-6: Fraction of all small satellites that failed due to the constant failure rate term of the PNZ modified Single-Weibull function (equation (35)).

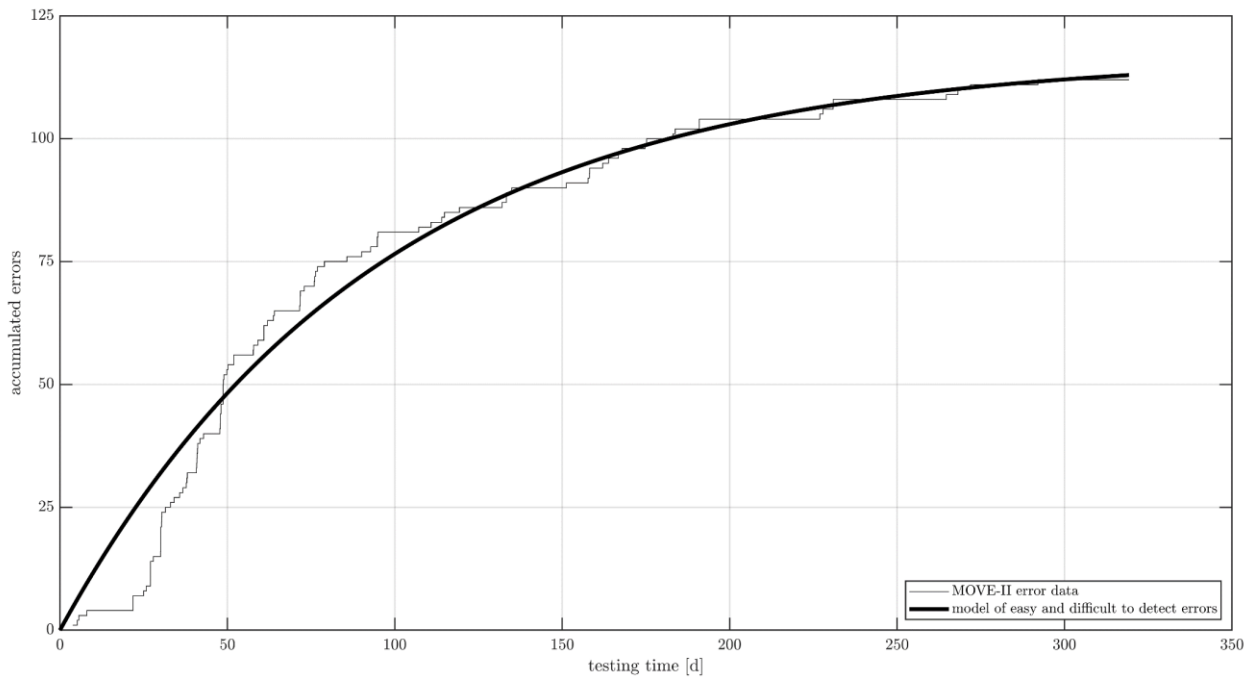


Figure 6-7: Fit of the Yamada & Osaki model to the critical failures of the space segment of MOVE-II.

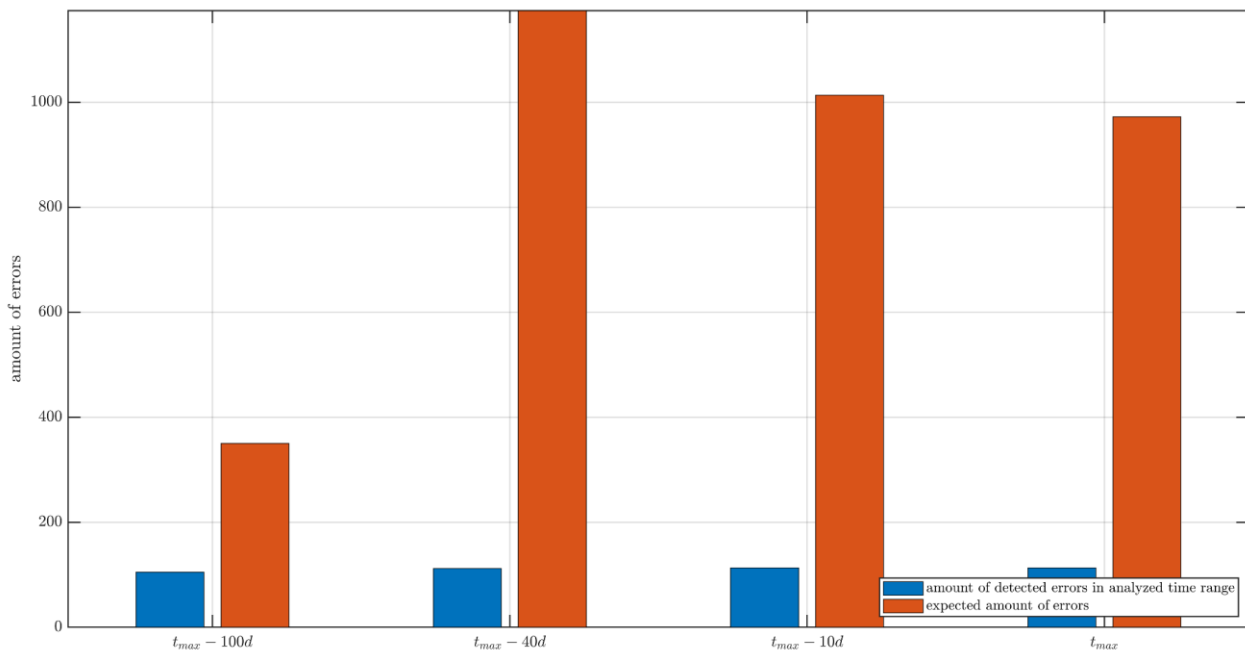


Figure 6-8: Stability of the prediction critical failures of the space segment of MOVE-II of the Yamada & Osaki model when going back to earlier points in time.

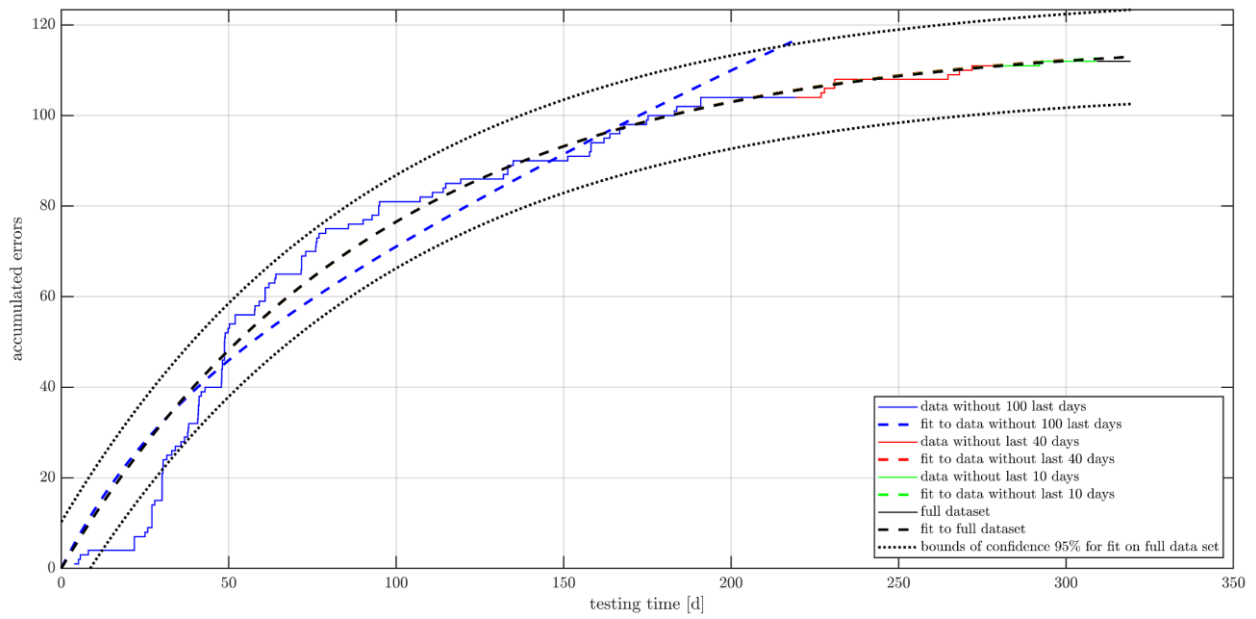


Figure 6-9: Prediction of critical failures of the space segment of MOVE-II by the Yamada & Osaki model using different time ranges. Blue depicts all data up to day 219, red all data up to day 279, green all data up to day 309 and black the full data set. The estimation of total errors in the system is increasing as the test proceeds.

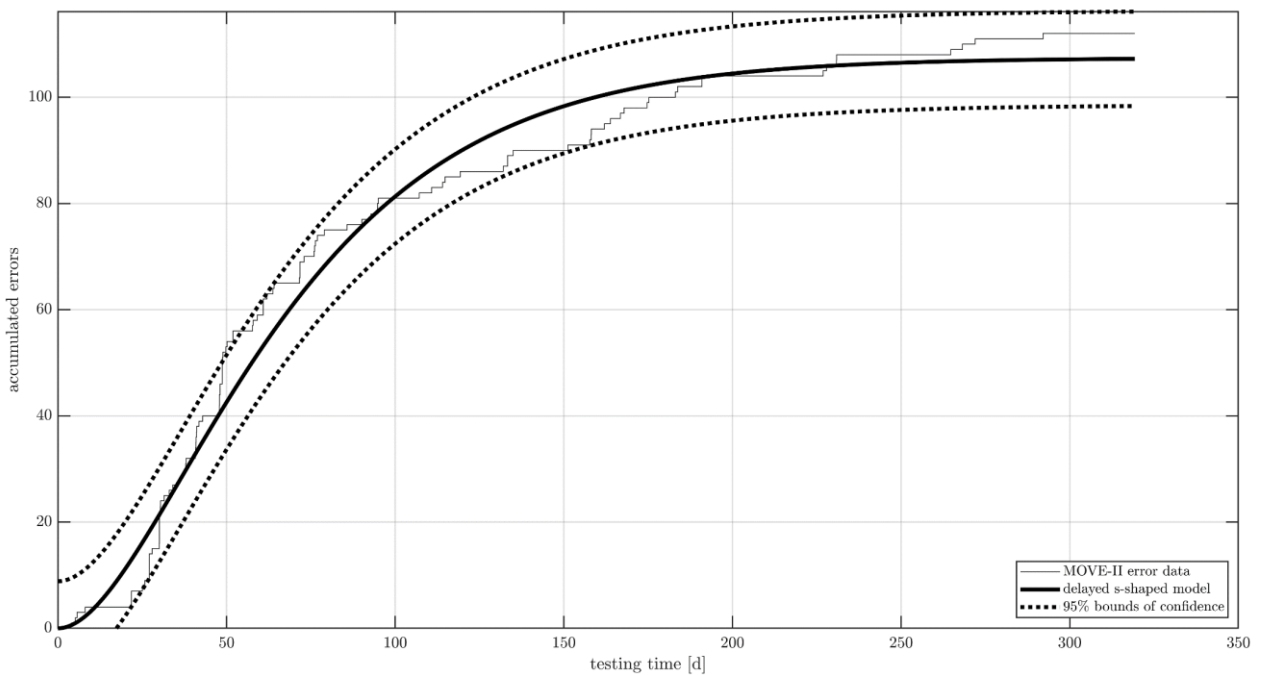


Figure 6-10: Fit of the delayed S-shaped software reliability growth model to the critical failures of the space segment of MOVE-II.

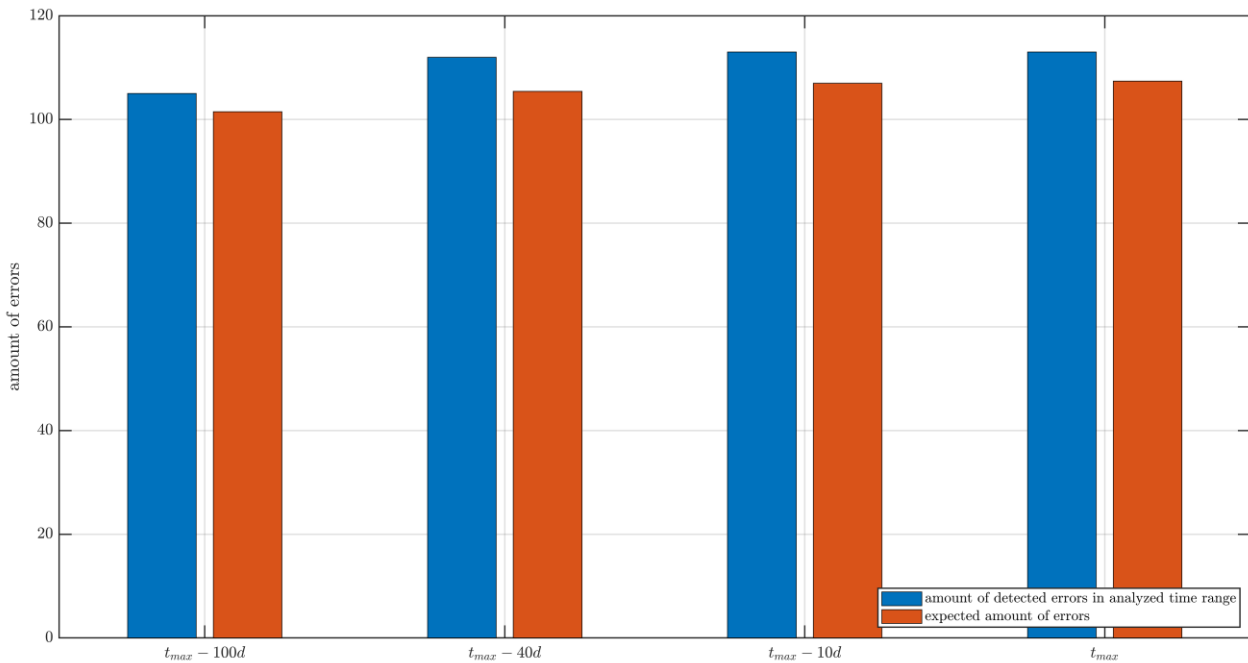


Figure 6-11: Stability of the prediction critical failures of the space segment of MOVE-II by the delayed S-shaped software reliability growth model when going back to earlier points in time.

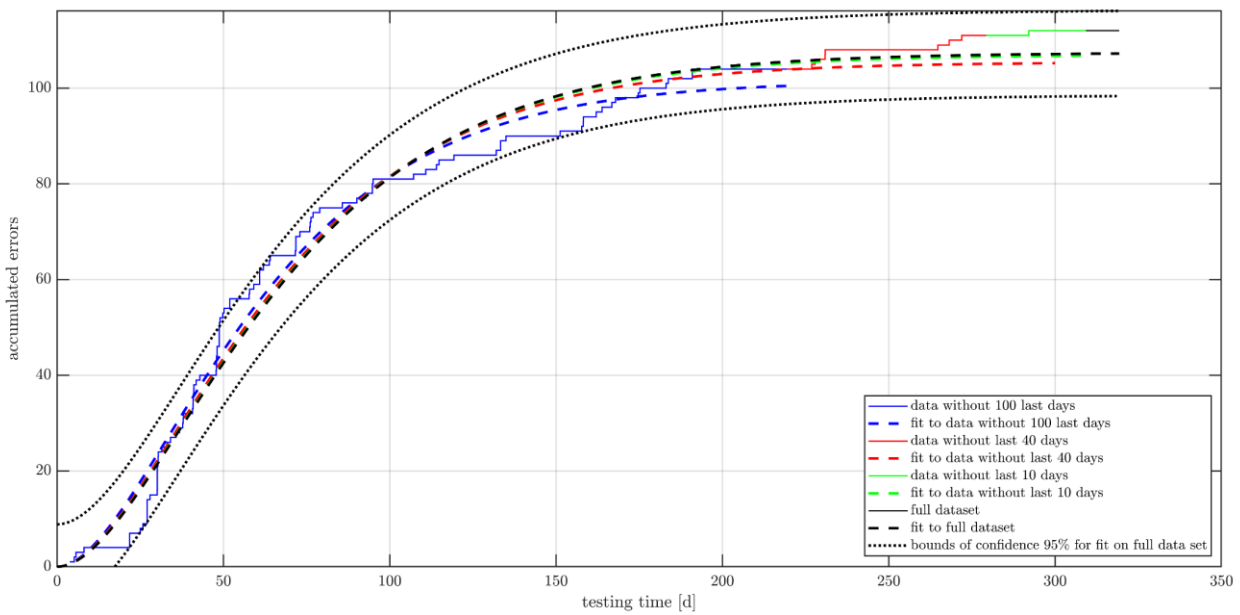


Figure 6-12: Prediction of critical failures of the space segment of MOVE-II by the delayed S-shaped software reliability growth model using different time ranges. Blue depicts all data up to day 219, red all data up to day 279, green all data up to day 309 and black the full data set. The estimation of total errors in the system is increasing as the test proceeds.

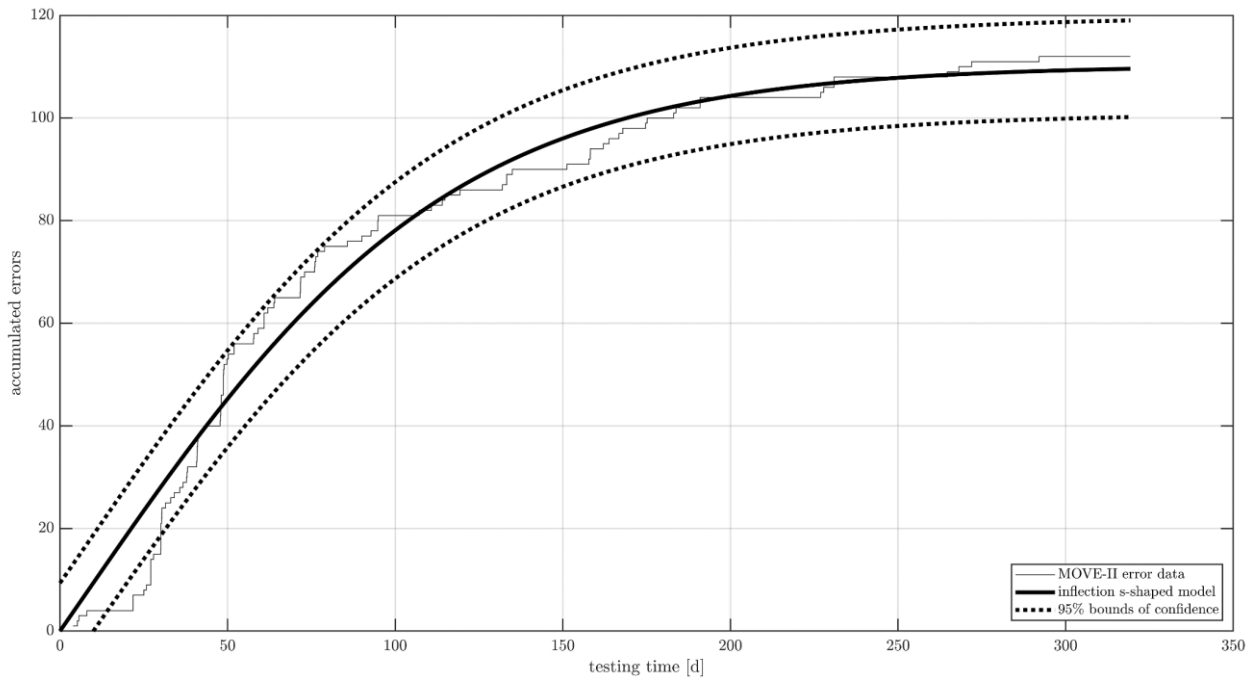


Figure 6-13: Fit of the inflection S-shaped software reliability growth model to the critical failures of the space segment of MOVE-II.

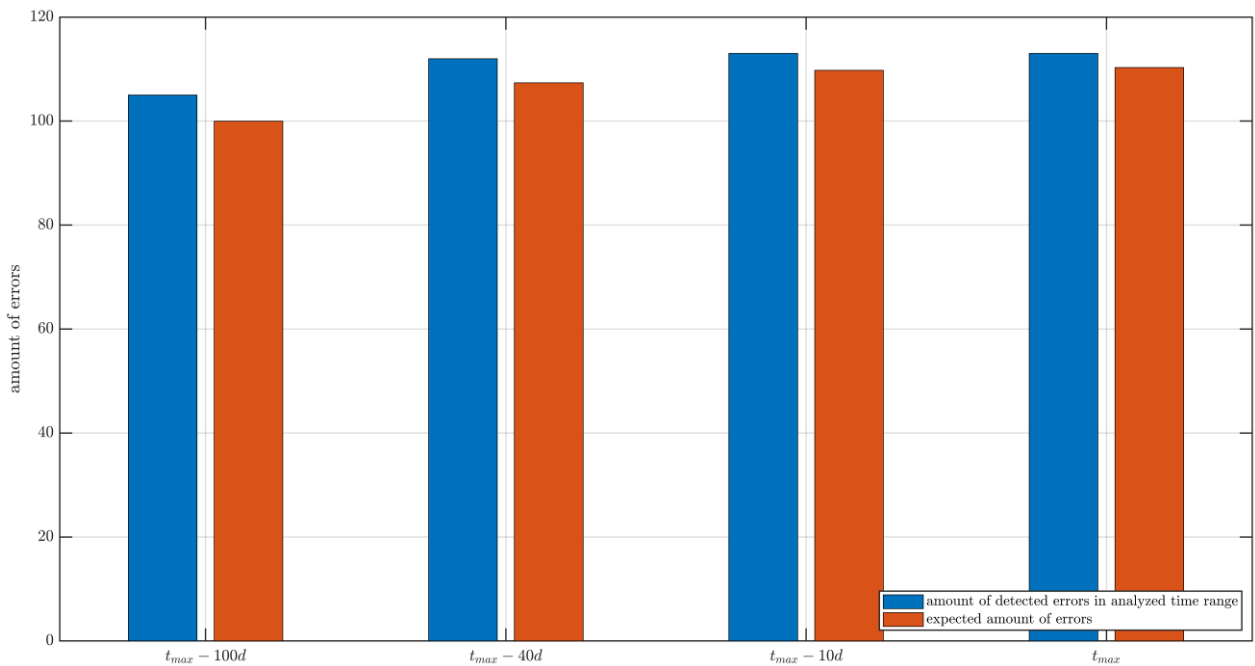


Figure 6-14: Stability of the prediction critical failures of the space segment of MOVE-II by the inflection S-shaped software reliability growth model when going back to earlier points in time.

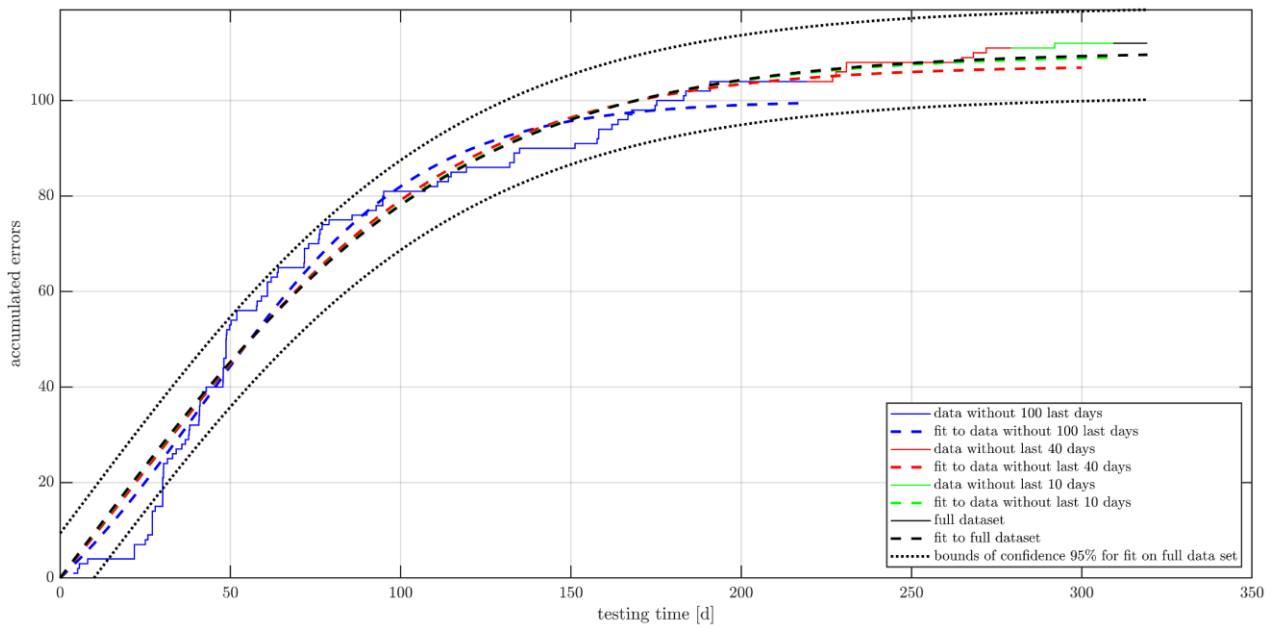


Figure 6-15: Prediction of critical failures of the space segment of MOVE-II by the inflection S-shaped software reliability growth model using different time ranges. Blue depicts all data up to day 219, red all data up to day 279, green all data up to day 309 and black the full data set. The estimation of total errors in the system is increasing as the test proceeds.

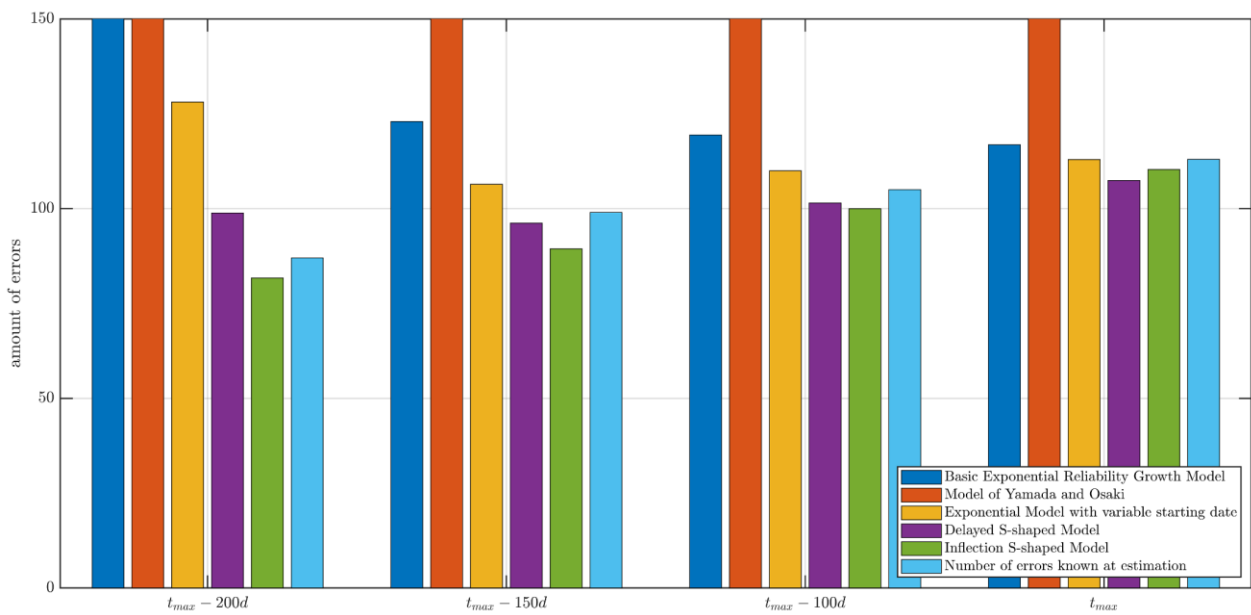


Figure 6-16: Comparison of the estimation of all models and the number of known errors at different points in time for all critical errors on the space segment of MOVE-II.

Table 6-8: Main techniques to promote teamwork in small satellite and CubeSat projects. Source: [14] (adapted)

1)	Cost goals. Project leaders accept cost containment (and schedule control) as a major goal. They communicate this goal to team members, who rank it as high in importance as the scientific and technical objectives of the project.
2)	Project scale. The spacecraft is small and the project team that develops and flies it is very small.
3)	Experienced and inexperienced personnel. Lacking formal safeguards, project leaders recruit experienced personnel who can recognize risks and resolve technical problems, and mix them with a larger number of inexperienced personnel.
4)	Technical discretion. Team members are allowed to control their own work, which includes the authority to make design changes and supervise contractors without outside interference.
5)	Protection. Team members are protected against red tape, annual budget caps, excessive outside review, external micromanagement, and other outside forces that threaten to limit their discretion.
6)	Stable funding. Within the total program cost cap, team leaders have the ability to spend funds at the most appropriate point in time. They receive funds when funds are needed.
7)	Co-location. Divided work packages and multicenter projects seriously hamper team effectiveness. Components of the spacecraft may be developed at different locations, but the central management team is located at one place.
8)	Multitasking. Multitasking is the process of moving team members from one job to another as the project matures, increasing the overall sense of responsibility and organizational memory.
9)	Hands-on activity. Team members learn about the spacecraft by working with actual hardware. They build and test the spacecraft themselves.
10)	Testing. Extensive testing, along with hands-on activity, allows team members to become intimately familiar with the intricacies of the spacecraft. It also reduces risk.
11)	Seamless management. Seamless management is the practice of using the same people on the project team from design through operations even as the nature of the work changes. This reinforces the principle of multitasking.
12)	Peer review. As a partial substitute for the checks lost by foregoing systems management, project workers present their plan to groups of their peers.
13)	Cancellation. Top executives are prepared to cancel any project where costs or schedule spiral out of control, and team members are told that this will happen.
14)	Risk-taking. Team members are encouraged to be creative and take calculated risks (but told not to fail).
15)	Risk management. Although project leaders do not emphasize formal systems management, they practice risk management techniques