

Projektspezifische Anpassung von Vorgehensmodellen: Feature-basiertes Tailoring

Georg Kalus

Institut für Informatik
der Technischen Universität München

**Projektspezifische Anpassung von
Vorgehensmodellen:
Feature-basiertes Tailoring**

Georg Kalus

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

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. Arndt Bode

Prüfer der Dissertation:

1. Univ.-Prof. Dr. Dr. h.c. Manfred Broy
2. Univ.-Prof. Dr. Alexander Knapp,
Universität Augsburg

Die Dissertation wurde am 31.01.2013 bei der Technischen Universität München
eingereicht und durch die Fakultät für Informatik am 31.05.2013 angenommen.

Kurzfassung

Vorgehensmodelle strukturieren in der Software- und Systementwicklung die komplexen Tätigkeiten und Abläufe eines Projekts. Die Vorgaben von Vorgehensmodellen sind in der Regel allgemeingültig und generisch beschrieben, um in einer möglichst großen Vielfalt von Projekten anwendbar zu sein. Für ein konkretes Projekt müssen die Vorgaben daher an die Situation des Projekts angepasst werden – ein Vorgang, der als *Tailoring* bezeichnet wird.

Projektcharakteristika, welche auf die projektspezifische Ausprägung eines Vorgehensmodells Einfluss nehmen, werden seit einigen Jahren diskutiert – allerdings sind diese meist nur informell beschrieben und ihre Wirkung auf die Vorgaben des Vorgehensmodells bleibt unklar bzw. ist bei jeder projektspezifischen Anpassung erneut und situationsspezifisch zu entscheiden. Dies erschwert zum einen die Durchführung des Tailorings und wirkt zum anderen hemmend auf gewünschte Effekte eines Vorgehensmodells, wie Nachvollziehbarkeit und Vergleichbarkeit der anwendenden Projekte.

In der vorliegenden Arbeit entwickeln wir den Ansatz des *feature-basierten Tailorings*. In dessen Zentrum steht ein formalisiertes *Konfigurationsmodell* zur Charakterisierung der Projektsituation anhand eines Katalogs von Tailoring-Kriterien. Die Projektcharakteristika im Konfigurationsmodell steuern über formal fundierte *Tailoring-Operationen* die *variablen Inhalte* des Vorgehensmodells und bestimmen damit Gestalt und Umfang einer projektspezifischen Ausprägung.

Das feature-basierte Tailoring ermöglicht die werkzeuggestützte Durchführung des Tailorings auch für einen nicht mit der inneren Struktur des Vorgehensmodells vertrauten Anwender. Wir diskutieren in der vorliegenden Arbeit Verfahren, die bei der werkzeuggestützten Durchführung des Tailorings hilfreich sind. Diese dienen einerseits der Schaffung von Transparenz für den Anwender, beispielsweise durch das Aufzeigen von Konsequenzen der Tailoring-Entscheidungen, und andererseits der Sicherstellung eines konsistenten Tailoring-Ergebnisses.

Zur Umsetzung und Anwendung des feature-basierten Tailorings in einem konkreten Vorgehensmodell beschreiben wir eine *Methode*. Diese wird unterstützt durch einen *Katalog von Tailoring-Kriterien*, welcher als Ausgangsbasis für die Entwicklung eines Konfigurationsmodells dient.

Die Ergebnisse dieser Arbeit werden in zwei Fallstudien auf konkrete Vorgehensmodelle angewendet. Die Anwendung dient dem Nachweis der Tragfähigkeit des feature-basierten Tailorings und demonstriert die Umsetzung anhand plastischer Beispiele.

Die Ergebnisse dieser Arbeit ermöglichen Prozessingenieuren die Entwicklung eines Vorgehensmodells, welches sich anhand einer Menge von Tailoring-Kriterien an die konkrete Situation eines Projekts anpassen lässt. Dabei werden die Tailoring-Kriterien, ihre hierarchische Organisation, Abhängigkeiten untereinander und ihre Wirkung auf die variablen Inhalte des Vorgehensmodells in einer Form beschrieben, welche die direkte Integration der Tailoring-Mechanismen in existierende Vorgehensmodelle erlaubt. Vorgehensmodell-Anwendern ermöglicht das feature-basierte Tailoring die werkzeuggestützte, nachvollziehbare und transparente Durchführung der Anpassung des Vorgehensmodells an eine konkrete Projektsituation.

Abstract

Process models give structure to the complex activities and procedures in software and systems development projects. The regulations in process models are usually described in a generic way to be applicable in a large variety of different projects. For a concrete project, the regulations have to be adjusted to the particular situation – an activity referred to as *tailoring*.

Project characteristics that have influence on a project-specific process model are being discussed for a number of years. However, the characteristics are usually described only informally and their impact on the process model often remains unclear or has to be decided individually and situation-specific for every tailoring. Not only does this complicate tailoring itself, it can also inhibit desired effects of process models, such as transparency and comparability of the applying projects.

In this thesis we are developing the approach of *feature-based tailoring*. At its heart lies a formalized *configuration model* for the description of a project's situation with a catalogue of tailoring criteria. The characteristics in the configuration model control the *variable contents* of the process model through formally defined *tailoring operations*. They thus determine the size and shape of the resulting project-specific process model.

Feature-based tailoring opens the door for tool support, allowing users without deep knowledge of the inner structure of the process model to perform tailoring. We are discussing techniques in this thesis that are helpful for the realization of tool support in tailoring. These on the one hand help to create transparency for the user, for example by making clear the consequences of tailoring decisions. On the other hand they help to ensure consistency in the tailoring result.

To realize feature-based tailoring in a concrete process model, we are describing a *method*. The method is supported by a *catalogue of tailoring criteria*. This catalogue can serve as the initial basis for the development of a concrete configuration model.

The results of this thesis are applied to real-world process models in two case studies. These applications serve as a proof of the capabilities of feature-based tailoring and are practical examples of its realization.

The results of this thesis allow process engineers the development of a process model that can be adapted to the situation of a project through a set of tailoring criteria. The criteria, their hierarchical organization and their interdependencies are described in a form that allows the mechanisms for tailoring to be directly integrated into existing process models. Feature-based tailoring allows users of a process model to adapt a process model to the concrete project situation by the help of tools and in a reproducible and transparent manner.

Danksagung

Besonders herzlich möchte ich mich bei Prof. Dr. Dr. h.c. Manfred Broy für das angenehme Umfeld am Lehrstuhl für Software & Systems Engineering bedanken. Die große Freiheit und Eigenverantwortung in der Arbeit gepaart, mit einem vertrauensvollen Umgang, boten einen idealen Nährboden für die Entwicklung eigener Ideen und schlussendlich für die Anfertigung dieser Dissertation. Besonders danken möchte ich auch meinem Zweitgutachter Prof. Dr. Alexander Knapp für unsere hilfreichen Diskussionen und die zahlreichen Anregungen zur Verbesserung der Arbeit.

Wesentliche Ideen der vorliegenden Dissertation sind im Rahmen einer Kooperation mit den Telekom Laboratories und der T-Systems International GmbH entstanden. Sehr dankbar bin ich Dr. Frank Marschall von T-Systems für die unzähligen und fruchtbaren Stunden der Tüftelei und für die Möglichkeit, meinen Beitrag zum SE Book leisten zu können. Dr. Philipp Offermann von den Telekom Laboratories ist es zu verdanken, dass unsere Arbeit dabei stets in zielführenden Bahnen verlief.

Großen Dank schulde ich all meinen langjährigen Kollegenfreunden am Lehrstuhl. Angesprochen können und sollen sich alle Kollegen am Lehrstuhl für Software & Systems Engineering fühlen und auch viele Kollegen am Institut über die Lehrstuhlgrenzen hinaus. Eine namentliche Erwähnung verdient Dr. Marco Kuhrmann. Über meine gesamte Zeit am Lehrstuhl (und schon davor) war er einer meiner engsten Vertrauten und Sparringspartner. Nicht nur hat unsere Zusammenarbeit mich zum Lehrstuhl geführt – er hat über die gesamte Zeit wesentlichen Anteil daran, dass ich jetzt diese Zeilen schreibe. Mein Dank gebührt auch Daniel Méndez-Fernández für den engen und fruchtbaren Austausch nicht nur über Arbeitsthemen. Danken möchte ich auch dem Glücksfall von Bürokollege Maximilian Irlbeck. Nie hätte ich mir einen besseren vorstellen können. Bedanken möchte ich mich auch bei der guten Seele des Lehrstuhls Silke Müller. Ihr Beitrag zum guten Klima am Lehrstuhl kann nicht hoch genug eingeschätzt werden.

Für die aufmerksame und akribische redaktionelle Durchsicht der Arbeit bedanke ich mich sehr herzlich bei Dr. Christian Kalus.

Zu guter Letzt möchte ich mich bei meinen langjährigen Freunden und bei meiner Familie, allen voran Margarete, Christian und Leni bedanken. Eure bedingungslose Unterstützung und Euer Rückhalt geben mir Kraft.

Mehr als die beste Freundin bist du, Mirjana. Keine andere Person hat so viel Anteil genommen an den Höhen und Tiefen während der Erstellung dieser Dissertation. Für die Geduld und Toleranz in den Tiefen und für mein Glück, die Höhen mit dir teilen zu können, bin ich dir für immer dankbar.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Wozu Tailoring?	3
1.2	Problemstellung	3
1.2.1	Hintergrund: Studie zur Anwendung eines Vorgehensmodells	5
1.3	Ziele der Arbeit	6
1.4	Lösungsansatz und Beitrag	6
1.4.1	Modell zur Charakterisierung von Projekten	7
1.4.2	Definierte Operationen des Tailorings	7
1.4.3	Werkzeugunterstützung	7
1.4.4	Kriterienkatalog zur Charakterisierung	7
1.4.5	Methode zur Anwendung	8
1.4.6	Beispiele	8
1.5	Forschungsmethode	8
1.6	Adressaten, Positionierung und Einschränkung	10
1.6.1	Anwendungsbereiche von Vorgehensmodellen	11
1.7	Aufbau dieser Arbeit	12
2	Ausgangslage und verwandte Arbeiten	15
2.1	Überblick über verwandte Arbeiten	16
2.2	Begriffe	17
2.2.1	Vorgehensmodell und Methode	17
2.2.2	Tailoring	22
2.3	Einordnung von Vorgehensmodellen	27
2.3.1	Kriterien und Eignungseinstufung	27
2.3.2	Aktivitäts- und Artefaktororientierung	29
2.3.3	Methodenneutralität	31
2.4	Charakterisierung von Projekten	33
2.4.1	Quantitatives Tailoring	33
2.4.2	Tailoring-Kriterien	34
2.4.3	Studie zur Anwendung eines Vorgehensmodells	38
2.4.4	Von Projekteigenschaften zum Vorgehensmodell	46
2.5	Verfahren zur Anpassung von Vorgehensmodellen und Methoden	51
2.5.1	Situational Method Engineering	51
2.5.2	Prozesslinienmodellierung	54

2.5.3	Tailoring-Ansätze	55
2.5.4	Projektspezifische Anpassung agiler Vorgehensweisen	56
2.6	Feature-Oriented Domain Analysis	57
2.6.1	Feature-Modelle	57
2.6.2	Analyse von Feature-Modellen	59
2.7	Zusammenfassung	63
3	Anpassungsfähigkeit von Vorgehensmodellen	65
3.1	Überblick	66
3.2	Reichhaltige Vorgehensmodelle	67
3.2.1	Das V-Modell XT	67
3.2.2	Das Eclipse Process Framework	72
3.2.3	HERMES - Die schweizerische Projektführungsmethode	76
3.2.4	SE Book	78
3.3	Leichtgewichtige Vorgehensweisen	81
3.4	Zusammenfassung	84
4	Struktur von Vorgehensmodellen	87
4.1	Überblick	88
4.2	Analyse: Das V-Modell XT Metamodell	89
4.2.1	Elemente und Beziehungen	89
4.2.2	Paketstruktur	92
4.2.3	Fazit	96
4.3	Analyse: das Metamodell SPEM	97
4.3.1	Paketstruktur	97
4.3.2	Method Content	99
4.3.3	Zusammenhang von Prozessstruktur und Methoden	99
4.3.4	Physikalische Repräsentation	101
4.3.5	Fazit	103
4.4	Analyse: Formalisierung leichtgewichtiger Methoden	103
4.4.1	Fazit	105
4.5	Ein einfaches Meta-Metamodell für Vorgehensmodelle	105
4.6	Tailoring neu formuliert	112
4.7	Zusammenfassung	113
5	Feature-basiertes Tailoring	115
5.1	Überblick	116
5.2	Charakterisierung eines Projekts	117
5.2.1	Modell zur Projektcharakterisierung	120
5.3	Operationen des Tailorings	124
5.4	Feature-basiertes Tailoring von Vorgehensmodellen	128
5.5	Sicherstellung von Konsistenz der projektspezifischen Ausprägung	133
5.5.1	Konstruktive Sicherstellung	134
5.5.2	Analytische Sicherstellung	136
5.6	Anwenderunterstützung bei der Durchführung des Tailorings	138
5.6.1	Reduktion des Konfigurationsaufwands	139
5.6.2	Ermittlung von Seiteneffekten	142
5.7	Zusammenfassung	143
6	Anwendung des feature-basierten Tailorings	145
6.1	Überblick	146
6.2	Rollen	146

6.3	Feature-basiertes Tailoring implementieren	147
6.3.1	Produkte	148
6.3.2	Ablauf	149
6.3.3	Anwendungsfall: Bedarf analysieren	151
6.3.4	Anwendungsfall: Variabilitätspunkte vereinbaren	152
6.3.5	Anwendungsfall: Kriterien vereinbaren	154
6.3.6	Anwendungsfall: Inhalte erstellen	156
6.3.7	Anwendungsfall: Werkzeuge implementieren	158
6.4	Vorgehensmodell projektspezifisch anpassen	160
6.4.1	Produkte	160
6.4.2	Vorgehen	160
6.5	Beispiele	161
6.6	Zusammenfassung	168
7	Evaluation	169
7.1	Überblick	170
7.2	Fallstudie: SE Book der T-Systems	170
7.2.1	Umfeld	170
7.2.2	Ziele	171
7.2.3	Studienaufbau	172
7.2.4	Ergebnisse	174
7.2.5	Diskussion	182
7.3	Fallstudie: V-Modell XT	183
7.3.1	Umfeld	183
7.3.2	Ziele	184
7.3.3	Studienaufbau	184
7.3.4	Ergebnisse	184
7.3.5	Diskussion	193
7.4	Experimente zur Evaluierung	194
7.4.1	Umfeld	194
7.4.2	Ziele und Forschungsfragen	195
7.4.3	Studienaufbau	196
7.4.4	Ergebnisse und Diskussion	198
7.4.5	Beeinträchtigung der Validität	203
7.5	Zusammenfassung	204
8	Zusammenfassung und Ergebnisse	205
8.1	Ausgangslage	206
8.2	Lösung und Ergebnisse	207
8.3	Ausblick und offene Forschungsfragen	209
A	Kriterienkatalog für das Tailoring von Vorgehensmodellen	213
A.1	Team	213
A.2	Inneres Umfeld	217
A.2.1	Management	217
A.2.2	Technische Umgebung	221
A.3	Äußeres Umfeld	224
A.3.1	Kunde	225
A.3.2	Endanwender	227
A.4	Projektgegenstand	228
B	Fragebogen zum Evaluierungsexperiment	235

Abbildungsverzeichnis

1.1	Ergebnisübersicht der Arbeit	6
1.2	Forschungsmethode der Arbeit	9
2.1	Struktur des Kapitels 2	16
2.2	Submodelle eines Vorgehensmodells	19
2.3	Modellierungsebenen nach MOF	20
2.4	Was ist Tailoring?	23
2.5	Lebenszyklus eines Vorgehensmodells	24
2.6	Tailoring in der Breite und in der Tiefe	27
2.7	Aktivitätsorientierung	30
2.8	Artefaktororientierung	30
2.9	Methodenneutrale und methodenspezifische Vorgehensmodelle	32
2.10	Kriterien zur Wahl einer Methode aus der Crystal Familie	37
2.11	Der S ³ Ansatz	38
2.12	Untersuchungsgegenstände der Studie	39
2.13	Ergebnis der Umfrage zum Einsatz von Vorgehensmodellen (1)	44
2.14	Ergebnis der Umfrage zum Einsatz von Vorgehensmodellen (2)	45
2.15	Ergebnis der Umfrage zum Einsatz von Vorgehensmodellen (3)	45
2.16	Prozess des Situational Method Engineering	52
2.17	Modell des Situational Method Engineering Ansatzes	54
2.18	Feature-Modell Beispiel	57
2.19	Feature-Modell Korrektheit	59
3.1	Struktur des Kapitels 3	66
3.2	Vorgehensmodell und Erweiterungsmodell im V-Modell XT	69
3.3	Metamodell zum Tailoring im V-Modell XT	69
3.4	V-Modell XT Projektassistent	71
3.5	Schritte des Tailorings im V-Modell XT	71
3.6	Kernelemente von EPF	73
3.7	EPF Composer	74
3.8	Konfiguration von Methodenbausteinen in SPEM	75
3.9	Tailoring in HERMES PowerUser	77
3.10	Tailoring im SE Book Process Suite Generator	79
3.11	Zusammenhang zwischen Labels und Modellelementen im SE Book	81
4.1	Struktur des Kapitels 4	88

4.2	V-Modell XT Inhalte als Baum im V-Modell XT Editor	89
4.3	Attribute referenzierbarer Elemente im V-Modell XT	90
4.4	Beispielausprägung der Attribute referenzierbarer Elemente	90
4.5	Beziehung als Attribut	91
4.6	Beispielausprägung einer Beziehung als Attribut	91
4.7	Beispielausprägung einer Beziehung als Element	92
4.8	Beziehung als Modellelement	92
4.9	Struktur eines Vorgehensbausteins	94
4.10	Struktur eines Produkts	95
4.11	Beziehungen rund um das Modellelement <i>Produkt</i>	96
4.12	Modellierungsebenen bei SPEM und UML	97
4.13	Paketstruktur des SPEM Metamodells	98
4.14	Zusammenhang zwischen Methodenbausteinen und Prozessstruktur	101
4.15	Überblick über die Pakete in MOF	102
4.16	Prozess-Daten Diagramm für Feature Driven Development	104
4.17	Kern des einfachen Meta-Metamodells	106
4.18	Variabilitätspunkte im Meta-Metamodell	107
4.19	$n : m$ Beziehung zwischen Produkten	111
5.1	Überblick über Feature-basiertes Tailoring	116
5.2	Struktur des Kapitels 5	117
5.3	Beispiele für Antwortmöglichkeiten	118
5.4	Metamodell für das Konfigurationsmodell zum Tailoring	120
5.5	Zusammenhang zwischen Metamodell und Notation für Feature- Modelle	122
5.6	Einfaches Meta-Metamodell für Vorgehensmodelle	127
5.7	Vereinfachtes Metamodell Feature-basiertes Tailoring	129
5.8	Komplettes Metamodell Feature-basiertes Tailoring	130
5.9	Rekapitulation: $n : m$ Beziehung zwischen Produkten	132
5.10	Beziehung zwischen Themen und Produkten im V-Modell XT	133
5.11	Struktureller Widerspruch in der Konfiguration	134
5.12	Ausdehnung variabler Modellanteile durch rekursive Nachverfolgung	135
5.13	Analytisches Verfahren zur Erkennung von Konflikten	136
5.14	Beispiel zur Illustration des analytischen Verfahrens zur Erkennung von Konflikten	137
5.15	Verfahren zur Bestimmung einer gültigen Konfiguration	140
6.1	Anwendungsfälle für die Umsetzung und Anwendung des feature- basierten Tailorings	146
6.2	Anwendungsfälle für die Implementierung des feature-basierten Tai- lorings	148
6.3	Prozess: Feature-basiertes Tailoring implementieren	150
6.4	Extreme bei der Wahl der Variabilitätspunkte	153
6.5	Metamodell zur Methodenintegration für das V-Modell XT	157
6.6	Beispielhafte Benutzeroberfläche zum Konfigurationsmodell	159
6.7	Vertragsart im SE Book Konfigurationsmodell	162
6.8	Auswirkung der Vertragsart auf weit entfernte Bereiche des Konfigu- rationsmodells	164
6.9	Kleinprojekt und Vorgehensmodellinhalte	165
6.10	Dokument des Kunden wird mehrfach eingebunden	167
7.1	Tailoring-Checkliste zum SE Book	171

7.2	Ausschnitt der Best-Practices für Cloud-Projekte	173
7.3	Struktur der Best-Practices für Cloud-Projekte	174
7.4	Feature-Baum im SE Book Metamodell	175
7.5	Cross-Tree Constraints im SE Book Metamodell	176
7.6	Zusammenhang zwischen Konfigurationsmodell und Prozessmodel- elementen	177
7.7	Überarbeitetes Tailoring im Process Suite Generator	179
7.8	PDE Editor für das SE Book	180
7.9	Hinweise für das Feature-Modell im SE Book PDE Editor	180
7.10	Abbildung der Cloud-Best-Practices mit Hilfe von Querbeziehungen	181
7.11	Abbildung der Cloud-Best-Practices durch Kopie der Elemente . . .	182
7.12	Metamodell zum Tailoring im V-Modell XT	183
7.13	Zusammenhang zwischen Projekteigenschaften und Inhaltselementen	185
7.14	Beispiel für die Abbildung des klassischen V-Modell XT Tailorings mit Querbeziehungen	189
7.15	Beispiel für die Abbildung des klassischen V-Modell XT Tailorings mit Kopien der Projektmerkmale	190
7.16	Migration des V-Modells auf Metamodell zum feature-basierten Tai- loring	191
7.17	Fiktives Beispiel umgesetzt mit feature-basiertem Tailoring	193
7.18	Experiment zur Evaluierung des feature-basierten Tailorings	197
7.19	Ergebnisse des Laborexperiments zum Tailoring im Process Suite Ge- nerator, dem EPF Composer und dem V-Modell XT Projektassistent .	199

Tabellenverzeichnis

2.1	Herausforderungen in Softwareentwicklungsprojekten	35
2.2	Strategien zur Bewältigung der Herausforderungen	36
2.3	Projektsteckbrief des Fragebogens	41
2.4	Fragebogen – Auszug Artefaktmodell	42
2.5	Ergebnis der Umfrage zum Einsatz des V-Modell XT	46
2.6	Referenz-Beiträge zur Ermittlung der Suchbegriffe	48
2.7	Aufbau der Literatursammlung	48
2.8	Statistik zur betrachteten Literatur	49
2.9	Tailoring-Maßnahmen in der Literatur	50
2.10	Feature-Modelle als aussagenlogischer Ausdruck	60
2.11	Aussagenlogischer Ausdruck in konjunktiver Normalform	62
3.1	Projektkategorien von HERMES	78
3.2	Projektgröße nach HERMES	78
3.3	Überblick über Inhalte leichtgewichtiger Methoden	83
4.1	Kernelemente eines SPEM Methodenbausteins	99
4.2	Kernklassen in SPEM <i>Process Structure</i>	100
4.3	V-Modell XT Metamodell als Instanz des Meta-Metamodells	110
5.1	Abbildung von Projektcharakteristika auf Konfigurationsmodell	125
5.2	Abbildung von Querbeziehungen auf Konfigurationsmodell	126
6.1	Vertragsart und Vorgehensmodellinhalte	163
6.2	Kunden-Vorgehensmodell und Vorgehensmodellinhalte	166
7.1	Anzahl der Produkte pro Vorgehensbaustein	187
7.2	klassisches V-Modell XT Tailoring mit Querbeziehungen	188
7.3	klassisches V-Modell XT Tailoring mit Kopien	189
7.4	Fiktive Projekteigenschaften bei AG-Projekt für IT-Dienstleister	192
7.5	Fiktive Projekteigenschaften bei AN-Projekt für IT-Dienstleister	192
7.6	Überblick über Projektcharakteristika	200
7.7	Einschätzung zur Durchführung des Tailorings	200
7.8	Ergebnisse zu Frage 7.a.	201
7.9	Ergebnisse zu Frage 7.b.	202

Die Erreichung von Projektzielen unter Einhaltung der Budget-, Qualitäts- und Zeitvorgaben bei der Softwareentwicklung hängt oft nicht so sehr von technologischen Einzelheiten ab, als viel mehr von der Qualität der Planung und der Organisation im Projekt [Sta04, BEJ06]. Während technologische Fragestellungen, z. B. der Programmierung, oft bis ins Detail verstanden sind, sind organisatorische Prozesse und Strukturen oft historisch gewachsen und nicht in ähnlichem Maße reflektiert.

Seit der Vorstellung eines einfachen Phasenmodells im Jahr 1956 [Ben83] und des Wasserfallmodells im Jahr 1970 [Roy70] haben Forschung und Praxis der ingenieurmäßigen Softwareentwicklung eine Vielzahl von Vorgehensmodellen zur Strukturierung von Softwareentwicklungsprojekten hervorgebracht. Seit Ende der neunziger Jahre des zwanzigsten Jahrhunderts wird vermehrt diskutiert, ob die entwickelten Methoden das adäquate Mittel sind und ob überhaupt die richtigen Fragen gestellt werden bzw. wurden. Stellvertretend steht das folgende Zitat von Alistair Cockburn (auf die Frage nach seiner Motivation zum Beitrag [Coc00a]):

Mostly to get rid of this aberration called software engineering. [...] engineering has given us a faulty vocabulary, a vocabulary that leads us in a direction that is not fruitful to a software project. It gives us questions like „Is the model accurate and complete?“ Much more interesting are „Do we have our goals aligned as a team?“ „Do we have the correct set of skills?“ and „What is the next move in this game, given that our competitor has just announced a new product?“

Alistair Cockburn in [Hig02]

Das Ergebnis dieser Diskussion sind sogenannte *agile* Vorgehensweisen und Praktiken, wie z. B. Scrum [Sch04] oder eXtreme Programming [Bec03], welche die kreativen Prozesse, die Interaktion zwischen Menschen im Team und Umgang mit Veränderung im Projekt in den Vordergrund stellen. Diese Vorgehensweisen decken meist nur den Mikrokosmos eines Projekts ab (lassen also insbesondere das organisatorische Umfeld außer Acht), sind in der Regel nur in kleinen Teams anwendbar und ihre Anwendung ist nicht frei von Risiken [SR03].

Es herrscht ein scheinbarer Widerspruch zwischen „traditionellen“, reichhaltigen Vorgehensmodellen und agilen Vorgehensweisen. Agile Vorgehen wurden nicht zuletzt als Gegenentwurf zu den als zu bürokratisch empfundenen schwergewichtigen Vorgehensmodellen entwickelt [Hig02, Boe02]. Dennoch:

I believe that both agile and plan-driven approaches have a responsible center and overinterpreting radical fringes. Although each approach has a home ground of project characteristics within which it performs very well, and much better than the other, outside each approach's home ground, a combined approach is feasible and preferable.

Barry Boehm in [Boe02]

Philippe Kruchten formuliert seine Gedanken zu dem scheinbaren Widerspruch so:

I would be very concerned that „rich“ rapidly means heavy, detailed, complicated; that „rich“ means taking the models we have and enriching them with additional concepts, more entities, more details, more relationships between them. [...] Can we have rich (in expressiveness and usefulness) and lean at the same time?

Philippe Kruchten in [Kru11]

Ein kombinierter Ansatz ist meist nicht nur zu bevorzugen, sondern geboten: agile Vorgehen beschäftigen sich kaum mit Schnittstellen zu Organisationseinheiten außerhalb des betrachteten Projekts. Zudem reichen oft schon vertragliche Rahmenbedingungen, um ein rein agiles Vorgehen zu erschweren. Darüber hinaus lassen sich durch eine explizite Definition – zumindest von Teilen – des Entwicklungsprozesses positive Effekte für die Projektdurchführung erzielen, sei es, weil dadurch Möglichkeiten zur Werkzeugunterstützung eröffnet werden [KKC09], oder um den Projektmitgliedern eine einheitliche Dokumentation der Prozesse zu bieten.

Reichhaltige Vorgehensmodelle blenden Aspekte wie Zusammenarbeit im Team, Verantwortung der einzelnen Entwickler und andere „weiche“ Aspekte oft weitgehend aus. So lange es sich um eine Unternehmung mit einer Größe handelt, die explizite Strukturen erfordert, *müssen* sich reichhaltige Vorgehensmodelle und agile Methoden ergänzen. Dass es hier immer noch zu Reibungen kommt, liegt unter anderem an folgenden Faktoren [Chr07]:

Verlust von Flexibilität Der durch ein Vorgehensmodell aufgespannte Rahmen wird als beengend empfunden, in welchem kreative Freiheit erstickt wird.

Mehraufwand Ein Vorgehensmodell verursacht unnötigen Mehraufwand, den man ohne das Vorgehensmodell nicht hätte.

Komplexität Ein Vorgehensmodell macht alles nur komplizierter. Anstatt beispielsweise eine Änderungsanfrage informell zu besprechen, müssen umständliche Prozesse eingehalten werden.

Wir sind anders Ein standardisiertes Vorgehen passt nicht zur Organisation oder zum Team. Man sei bisher auch ohne ausgekommen.

Das Produkt ist wichtiger als Buchhaltung Die Entwicklung steht im Vordergrund – da bleibt keine Zeit für Dokumentation, Spezifikation, etc.

Transparenz Ein Vorgehensmodell und die damit einhergehende Dokumentation kann Schwachstellen, Fehler und Misstände im Projekt und in der Arbeit Einzelner aufdecken.

Um diesen Vorbehalten zu begegnen, ist es erforderlich, dass schwergewichtige Vorgehensmodelle flexibler in der Anwendung werden und ihr bürokratischer Überhang so weit wie möglich reduziert wird. Agile Gegenentwürfe sind kein Ersatz, sondern als *Ergänzung* und Bereicherung des Methodenrepertoires zu verstehen. Ziel muss es dementsprechend sein, die Brücke zwischen beiden Philosophien zu schlagen und eine sinnvolle Integration zu ermöglichen.

Der Hebel ist bei Vorgehensmodellen selbst anzusetzen: Vorgehensmodelle müssen in geeigneter Weise an das vorliegende Projekt anpassbar sein. Die Anpassung an projektspezifische Umstände wird gemeinhin als *Tailoring* bezeichnet.

1.1 Wozu Tailoring?

Ein Vorgehensmodell stellt immer eine idealisierte und verallgemeinerte Beschreibung eines konkreten Softwareentwicklungsprojekts dar. Insofern ist eine Interpretation, Ausgestaltung und Abweichung innerhalb gewisser Leitplanken in der Realität der Projektdurchführung unumgänglich. Wird die Abweichung aber zu groß, können sich die durch den Einsatz eines Vorgehensmodells erhofften positiven Effekte, wie z. B. Vollständigkeit und Qualität der zu liefernden Ergebnisse, Eindämmung der Kosten und Nachvollziehbarkeit [Gna05], nicht entfalten. Neben dem Ausbleiben positiver Effekte kann sich ein „Ausbrechen aus den Leitplanken“ auch negativ auswirken, wenn z. B. die durch das Vorgehensmodell vorgeschriebenen Ergebnisse Grundlage für ein artefaktbasiertes Controlling sind [MF11]. Ein nicht oder unvollständig erstelltes Arbeitsergebnis würde sich direkt in den Statistiken und Kennzahlen zum Projekt niederschlagen, auch wenn es möglicherweise bewusst und aus sachlich vernünftigen Gründen nicht oder unvollständig (im Sinne des Vorgehensmodells) erstellt wurde.

Das Vorgehensmodell sollte keinen unnötigen bürokratischen Mehraufwand im Projekt verursachen. Die ideale Vorgehensweise laut [Coc04] ist „barely sufficient“ – also gerade ausreichend aber auch nicht weniger. Ist ein Vorgehensmodell zu schlank und tolerant, was beispielsweise die Dokumentation im Projekt oder die Vorgaben für Kommunikation (innerhalb des Projekts und nach außen, z. B. zum Kunden und zur Geschäftsleitung) angeht, besteht ebenfalls das Risiko, dass die oben erwähnten positiven Effekte eines Vorgehensmodells ihre Wirkung nicht entfalten können. Da per Definition jedes Projekt einzigartig ist (vergleiche Definition 2), bleibt nur das Maßschneidern des Vorgehensmodells an die Charakteristika des vorliegenden Projekts.

1.2 Problemstellung

Die Notwendigkeit der Anpassung einer Vorgehensweise an die Umstände eines Projekts ist allgemein anerkannt [Bro87, FRO03, Bri96, tV97, Coc00b, XR08, Kuh06, OMG08, Kuh07].

Große Übereinstimmung herrscht in der Literatur weiterhin darüber, dass die Anpassung basierend auf einer Projektcharakterisierung – im Situational Method Engineering *Situation* des Projekts genannt – erfolgen sollte. Dennoch können bei existierenden Ansätzen zum Tailoring folgende Unzulänglichkeiten ausgemacht werden:

1.2 Problemstellung

Problem: Für die Charakterisierung der Projektsituation gibt es kein *Modell* und die Charakterisierung erfolgt in vielen Ansätzen *getrennt* und *unabhängig* vom betroffenen Vorgehensmodell. Zwar gibt es zahlreiche Vorschläge und Hinweise, welche Faktoren auf die Projektsituation und damit auf das Vorgehensmodell einwirken können, diese sind aber informell beschrieben und Wechselwirkungen zwischen Situationsfaktoren bleiben unklar. „The main problems with contingency frameworks, such as Multiview, is that they lack a theoretical foundation for the environment factors and above all do not supply guidelines for the selection of method fragments on the basis of these factors“ [tV97].

Problem: In informellen Ansätzen zum Tailoring ist unklar, welche Wirkung die Charakterisierung des Projekts auf das Vorgehensmodell haben soll. Auch hierzu gibt es (beispielsweise in [XR08]) informelle Hinweise – die tatsächliche Umsetzung der Hinweise in einem konkreten Vorgehensmodell bleibt unbeantwortet. Was genau bedeutet es, wenn aufgrund einer kleinen Teamgröße der Umfang des Vorgehensmodells reduziert werden sollte? Was genau ändert sich im Vorgehensmodell, wenn der Kunde im Projekt nicht verfügbar ist? Welche Auswirkungen hat die Verwendung einer bestimmten Implementierungstechnologie?

In [Har97] wird ein Abgleich der Situationsfaktoren mit Methodenfragmenten vorgeschlagen. In einer SQL-artigen Anfragesprache sollten diejenigen Methodenfragmente gewählt werden, die auf die Projektsituation zutreffen. Jedoch ist erstens keine Umsetzung dieser Anfragesprache auf einer Methodenbibliothek bekannt und zweitens bliebe die Bildung einer kompletten und konsistenten projektspezifischen Ausprägung des Vorgehensmodells aus den zutreffenden Methodenfragmenten bei diesem Vorschlag dem Prozessingenieur überlassen¹.

Problem: Wenn ein *Modell* der Projektcharakterisierung mit *definierter Wirkung* auf die Inhalte der projektspezifischen Ausprägung des Vorgehensmodells vorhanden ist – wie im V-Modell XT (siehe Abschnitt 3.2.1), in Teilen in HERMES (siehe Abschnitt 3.2.3) und im SE Book (vergleiche Abschnitt 3.2.4) – bleibt dieses so rudimentär, dass

- nur eine grobe Einstufung des Projekts anhand weniger Kriterien möglich ist und das Modell nicht für eine umfassende Charakterisierung skaliert,
- dadurch ein feingranulares Tailoring, beispielsweise zur Einbindung spezifischer Methoden für eine spezielle Projektsituation nicht möglich ist,
- Wechselwirkungen zwischen Charakteristika nicht darstellbar sind und
- es inflexibel ist, was die Wirkung der Charakterisierung auf die Inhalte der projektspezifischen Ausprägung des Vorgehensmodells angeht.

Die unzureichenden Möglichkeiten der projektspezifischen Anpassung von Vorgehensmodellen führen dazu, dass eine Diskrepanz zwischen den Vorgaben des Vorgehensmodells und der gelebten Realität im Projekt festgestellt werden

¹ Wenn es eine Umsetzung des Abgleichs zwischen Situationsfaktoren und Methodenfragmenten gäbe, wäre die Durchführung dieser Tätigkeit aufgrund der erforderlichen Kenntnis der Struktur der Fragmente und des gesamten Vorgehensmodells kaum einem *Anwender des Vorgehensmodells* – beispielsweise dem Projektleiter – zuzumuten.

kann. In [LL04] wird dieses Phänomen als *method usage tension* bezeichnet: es gibt eine Spannung zwischen der *method-in-concept* und der *method-in-action*.

1.2.1 Hintergrund: Studie zur Anwendung eines Vorgehensmodells

Um die Anwendung, Akzeptanz und den Bedarf für Anpassung von Vorgehensmodellen zu ermitteln, haben wir bei der T-Systems International GmbH eine Studie zur Verwendung des dort hausintern verwendeten Vorgehensmodells *SE Book* durchgeführt. Eine detaillierte Beschreibung der Studie findet sich in Abschnitt 2.4.3.

Das Ziel der Studie war, folgende Fragen zu beantworten: Wie wird das Vorgehensmodell eingesetzt? Welche Teile des Vorgehensmodells werden (überwiegend) eingesetzt? Welche Aspekte des Projektalltags sind zur Zeit nur unzureichend abgedeckt? Gibt es eine Korrelation zwischen Projektcharakteristika und der Anwendung des Vorgehensmodells? Lässt sich daraus der Bedarf für Kriterien zur Anpassung ableiten?

Um diese Fragen zu beantworten, wurden alle zur Verfügung stehenden Quellen ausgewertet: Das Problem- und Änderungsmanagement zum Vorgehensmodell, die Zugriffsstatistiken der Webseite, eine an Anwender des Vorgehensmodells gerichtete Umfrage und persönliche Gespräche mit Teilnehmern derselben.

Zentrale Ergebnisse der Studie sind:

- Die Dokumentation des Vorgehensmodells ist unwichtig im Vergleich zu den rund um das Vorgehensmodell zur Verfügung gestellten Hilfsmitteln, wie z. B. Vorlagen für Arbeitsergebnisse, Checklisten, usw.
- Das Vorgehensmodell wird pragmatisch gelebt. Wenn das Vorgehensmodell nicht zur Realität des Projekts passt, werden Inhalte des Vorgehensmodells ignoriert oder fehlende Inhalte durch individuelle Vorgaben ergänzt. Diese Situation könnte durch ein besser auf die Umstände des Projekts angepasstes Vorgehensmodell verbessert werden.
- Die Anwender wünschen sich mehr Flexibilität in der Breite des Vorgehensmodells, beispielsweise sind in Beratungsprojekten oft nur ausgewählte Phasen der Projektdurchführung von Bedeutung.
- Die Anwender des Vorgehensmodells haben Bedarf hinsichtlich der Tiefe der vom Vorgehensmodell zur Verfügung gestellten Inhalte. Beispielsweise werden Hinweise und Richtlinien für bestimmte Implementierungstechnologien, für Beratungsprojekte und für den Umgang mit Unterauftragnehmern gewünscht.

Problem: Deutlich bestätigt durch unsere Studie wurde das Phänomen der *Method Usage Tension* [LL04]: Die Anwender des *SE Book* ignorieren diejenigen Teile des Vorgehensmodells, die nicht auf die Projektsituation zutreffen. Nicht auf die Situation zutreffende Inhalte blähen die Vorgaben des Vorgehensmodells auf und erschweren das Auffinden der relevanten Inhalte. Gleichzeitig wünschen sich die Anwender spezifische Methoden und Vorgaben im Vorgehensmodell, beispielsweise zur Verwendung bestimmter Implementierungstechnologien.

1.3 Ziele der Arbeit

Übergeordnetes Ziel der vorliegenden Dissertation ist die Reduktion der *Method Usage Tension* durch Entwicklung von Mechanismen, die eine Ableitung einer projektspezifischen Ausprägung eines Vorgehensmodells basierend auf einem Konfigurationsmodell zur Projektcharakterisierung ermöglichen.

Die projektspezifische Anpassung soll von einem nicht bis ins letzte Detail mit den inneren Strukturen des Vorgehensmodells vertrauten *Anwender des Vorgehensmodells* (üblicherweise dem Projektleiter zu Projektbeginn) durchführbar sein. Zu diesem Zweck muss das Vorgehensmodell für den *Entwickler des Vorgehensmodells* Mittel zur Beschreibung der *Projektcharakteristika* und ihrer *Wechselwirkung* beinhalten. Weiterhin muss es dem Entwickler des Vorgehensmodells möglich sein, über *definierte Operationen* die Auswirkung der Projektcharakterisierung auf die *variablen Anteile* des Vorgehensmodells auszudrücken. Ziel der vorliegenden Arbeit ist die Entwicklung dieser Mittel.

1.4 Lösungsansatz und Beitrag

Wir entwickeln in der vorliegenden Dissertation den Ansatz des feature-basierten Tailorings. Dieser ermöglicht die Charakterisierung der Projektsituation anhand eines festen Katalogs von Tailoring-Kriterien. Der Katalog ist durch ein *Konfigurationsmodell* beschrieben, welches die hierarchische Organisation der Kriterien und die Darstellung von Wechselwirkungen zwischen den Kriterien ermöglicht. Das Konfigurationsmodell steht über definierte *Tailoring-Operationen* mit den *variablen Inhalten* des Vorgehensmodells in Beziehung. Über die Tailoring-Operationen wird ausgedrückt, welche Wirkung die Belegung eines Kriteriums auf die Inhalte der projektspezifischen Ausprägung des Vorgehensmodells haben soll.

Abbildung 1.1 gibt einen Überblick über die Ergebnisse dieser Arbeit. In den folgenden Abschnitten betrachten wir die einzelnen Beiträge näher.

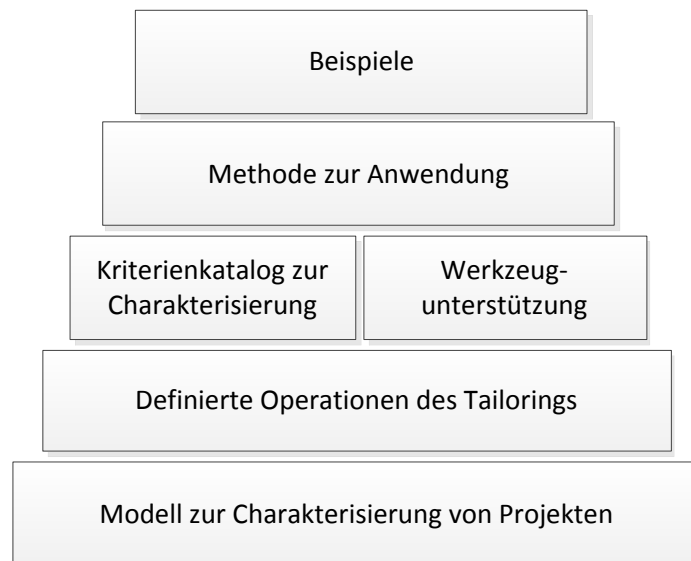


Abbildung 1.1: Ergebnisübersicht der Arbeit

1.4.1 Modell zur Charakterisierung von Projekten

Wir entwickeln ein *Konfigurationsmodell*, welches die Beschreibung der Projektcharakteristika erlaubt. Im Gegensatz zu existierenden Ansätzen (vergleiche Abschnitt 1.2) ist das Modell zur Situationsbeschreibung des Projekts hinsichtlich Umfang und Detaillierungsgrad nicht beschränkt. Weiterhin erlaubt das Konfigurationsmodell die Darstellung von Wechselwirkungen zwischen Kriterien, beispielsweise Abhängigkeiten und gegenseitigen Ausschluss. Bei der Entwicklung des Konfigurationsmodells orientieren wir uns an den Techniken der Produktlinienmodellierung – insbesondere der Feature-Modelle. Durch die definierten Operationen des Tailorings besteht ein klarer und wohldefinierter Zusammenhang zwischen dem Konfigurationsmodell und den von der Konfiguration betroffenen Vorgehensmodellinhalten. Das ist ein wesentlicher Unterschied zu alternativen Ansätzen, bei denen zwischen Tailoring und den betroffenen Inhalten des Vorgehensmodells keine klare Beziehung besteht.

1.4.2 Definierte Operationen des Tailorings

Die Operationen des Tailorings setzen das Konfigurationsmodell und die von der Konfiguration betroffenen Inhalte des Vorgehensmodells in Beziehung.

In Kapitel 4 kondensieren wir aus der Betrachtung populärer Metamodelle für Vorgehensmodelle ein einfaches Meta-Metamodell für Vorgehensmodelle, welches die Auszeichnung der variablen und statischen Inhalte in einem Vorgehensmodell erlaubt und damit als Grundlage für die Verknüpfung von Projektcharakterisierung und Inhalten dient.

Zur Verknüpfung der im Meta-Metamodell für Vorgehensmodelle definierten *Variabilitätspunkte* und der Kriterien zur Projektcharakterisierung entwickeln wir *Tailoring-Operationen*, die mit einer eindeutigen Semantik die Wirkung der Charakterisierung auf die Inhalte der projektspezifischen Ausprägung des Vorgehensmodells beschreiben.

1.4.3 Werkzeugunterstützung

Die Durchführung der projektspezifischen Anpassung mit Hilfe des feature-basierten Tailorings soll einem Anwender des Vorgehensmodells – insbesondere dem Projektleiter zu Projektbeginn – möglich sein. Um dieses Ziel zu erreichen, entwickeln wir in der vorliegenden Dissertation Verfahren zur Realisierung eines Werkzeugs zur Durchführung des Tailorings. Daneben stellen wir auch einige Verfahren vor, die sich in der praktischen Anwendung des feature-basierten Tailorings – beispielsweise im Rahmen der Fallstudien (siehe Kapitel 7) – als nützlich und hilfreich erwiesen haben.

1.4.4 Kriterienkatalog zur Charakterisierung

Damit der in dieser Arbeit entwickelte Ansatz des feature-basierten Tailorings nicht abstrakt bleibt und als Anregung und Hilfe bei der Anwendung haben wir in Anhang A einen Kriterienkatalog zur Charakterisierung von Softwareentwicklungsprojekten zusammengestellt. Der Katalog beinhaltet Kriterien, die in der Literatur und in existierenden Vorgehensmodellen als maßgeblich angesehen werden. Zu jedem Kriterium sind mögliche Wechselwirkungen und Querbezüge zu anderen Kriterien angedeutet. Weiterhin sind zu jedem Kriterium bei-

1.5 Forschungsmethode

spielhaft mögliche Wirkungen auf die Inhalte einer projektspezifischen Ausprägung des Vorgehensmodells skizziert. Die Ausgestaltung der Querbezüge und der Wirkung auf die Inhalte des Vorgehensmodells hängen bei der Umsetzung des feature-basierten Tailorings natürlich stark vom Anwendungsfall ab.

1.4.5 Methode zur Anwendung

Das feature-basierte Tailoring entwickeln wir in der vorliegenden Arbeit unabhängig von einem konkreten Vorgehensmodell. Konkrete Vorgehensmodelle verwenden wir, um die Konzepte und Verfahren zu illustrieren und zu überprüfen.

Um die Umsetzung und Anwendung in einem konkreten Vorgehensmodell zu ermöglichen, entwickeln wir eine Methode zum feature-basierten Tailoring – einerseits für die Rolle des Prozessingenieurs und andererseits für die Rolle des Vorgehensmodell-Anwenders.

Die Methode für den Prozessingenieur hilft durch Beschreibung von Anwendungsfällen und durch Vereinbarung eines exemplarischen Prozesses bei der Umsetzung des feature-basierten Tailorings. Insbesondere gibt die Methode Hinweise zur Definition des Konfigurationsmodells, der Variabilitätspunkte und zur Entwicklung des Werkzeugs zur Durchführung des feature-basierten Tailorings.

Die Methode für den Anwender umfasst nur den Anwendungsfall *Vorgehensmodell projektspezifisch anpassen* und beinhaltet Hinweise für den Projektleiter bei der Durchführung des feature-basierten Tailorings zur Erlangung einer projektspezifischen Ausprägung des Vorgehensmodells.

1.4.6 Beispiele

Zur plastischen Illustration aller anderen Beiträge dieser Arbeit, zur Sicherstellung der Umsetzbarkeit und zur Prüfung der Validität wenden wir das feature-basierte Tailoring in mehreren konkreten Beispielen an. Da den Beispielen jeweils ein echtes Vorgehensmodell zugrunde liegt, machen sie anschaulich, wie die allgemein und abstrakt beschriebenen Konzepte der Arbeit in der Realität umgesetzt werden (können).

1.5 Forschungsmethode

Um die Validität der Ergebnisse sicherzustellen, beschreiben wir im Folgenden die in dieser Dissertation zur Anwendung kommende Forschungsmethode. Ein Überblick über das Vorgehen dieser Arbeit ist in Abbildung 1.2 dargestellt.

Methodisch orientieren wir uns an dem gestaltungsorientierten (designwissenschaftlichen) Forschungsparadigma nach [sBFea10, Fra10]. Im Gegensatz zu verhaltenswissenschaftlichen Ansätzen zielt dieser Ansatz darauf ab, durch innovative Artefakte wichtige, relevante Probleme einer Organisation zu lösen [WH06]. Im Folgenden gehen wir kurz auf die vier Phasen des hier verfolgten Ansatzes ein:

Analyse: Die grundsätzliche Problemstellung haben wir bereits dargestellt. Der Bedarf für projektspezifische Anpassung eines Vorgehensmodells wird ohnehin nicht in Frage gestellt [Bro87, FRO03, Bri96, tV97, Coc00b, XR08, Kuh06,

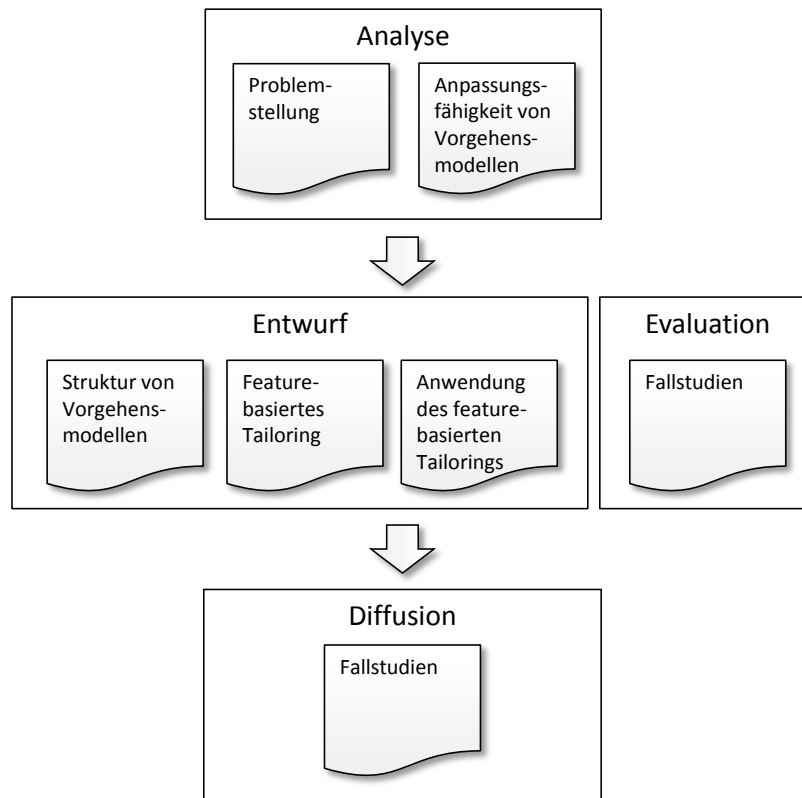


Abbildung 1.2: Forschungsmethode der Arbeit

OMG08, Kuh07]. In der bereits skizzierten Studie zur Ermittlung des Bedarfs für projektspezifische Anpassung untersuchen wir, welche Anforderungen an projektspezifische Anpassung Anwender eines Vorgehensmodells haben. Diese Studie wird in Abschnitt 2.4.3 ausführlicher beschrieben und stellt – zusammen mit verwandten Studien – eine wesentliche Motivation für die vorliegende Arbeit dar. Zusätzliche Motivation ergibt sich aus der Analyse populärer Vorgehensmodelle hinsichtlich ihrer Fähigkeit zur projektspezifischen Anpassung in Kapitel 3. Diese Analyse bestätigt und detailliert die Problemstellung.

Entwurf: Basierend auf der Problemanalyse entwickeln wir in der vorliegenden Arbeit einen Ansatz zur projektspezifischen Anpassung von Vorgehensmodellen basierend auf Feature-Modellen. Der Ansatz eignet sich für Vorgehensmodelle, die auf einem Metamodell basieren und somit in ihrer Struktur als Graph darstellbar sind. Wir analysieren deshalb in Kapitel 4 die Struktur bekannter Vorgehensmodelle und leiten daraus ein einfaches Meta-Metamodell für Vorgehensmodelle ab.

In Kapitel 5 entwickeln wir den Ansatz des feature-basierten Tailorings. Wir entwickeln ein Metamodell zum feature-basierten Tailoring, diskutieren die Semantik desselben und skizzieren, wie das Tailoring basierend auf dem entwickelten Metamodell in Werkzeugen umgesetzt werden kann. Das feature-basierte Tailoring bietet einige wünschenswerte Eigenschaften – beispielsweise kann es dabei helfen, die Erstellung einer strukturell inkonsistenten Ausprägung des Vorgehensmodells als Resultat des Tailorings zu vermeiden. Die Überprüfung dieser Eigenschaften ist ebenfalls Bestandteil des Kapitels 5.

Teil des Ansatzes zum feature-basierten Tailoring ist die Methode seiner An-

1.6 Adressaten, Positionierung und Einschränkung

wendung. In Kapitel 6 entwickeln wir eine derartige Methode. Diese zeigt für den Prozessingenieur Schritte auf, die zur Umsetzung des feature-basierten Tailorings in einem eigenen Vorgehensmodell nötig sind. Für den Anwender eines Vorgehensmodells führen wir ebenfalls aus, wie er ein Vorgehensmodell mit Hilfe des feature-basierten Tailorings an die Situation eines konkreten Projekts anpasst.

Evaluation: Die Evaluation der Ergebnisse dieser Arbeit erfolgt qualitativ anhand zweier Fallstudien in Kapitel 7, bei denen wir das feature-basierte Tailoring in zwei realen Vorgehensmodellen umsetzen. Durch die Umsetzung zeigen wir erstens die tatsächliche Anwendbarkeit und gegenüber dem Ist-Stand insbesondere auch den Mehrwert und Grenzen. Weiterhin beschreiben wir in Kapitel 7 ein kontrolliertes Experiment, bei dem die Stärken und Schwächen des Tailorings im Eclipse Process Framework, beim V-Modell XT und beim SE Book aus Sicht eines Vorgehensmodell-Anwenders analysiert werden.

Diffusion: Das Ziel jeder gestaltungsorientierten Forschung muss der Transfer der Ergebnisse in die Praxis sein [sBFea10]. Die bei T-Systems International GmbH durchgeführte Fallstudie (siehe Abschnitt 7.2) befasst sich mit dem in diesem Unternehmen hausintern eingesetzten Vorgehensmodell *SE Book*. Die Erweiterungen des Tailorings in diesem Vorgehensmodell basierend auf den Ergebnissen der vorliegenden Arbeit befinden sich im industriellen Einsatz.

Mit der Fallstudie zum V-Modell XT (siehe Abschnitt 7.3) und mit der Methode zur Umsetzung und Anwendung in Kapitel 6 ist mindestens der Grundstein für die weitere Diffusion des feature-basierten Tailorings gelegt.

1.6 Adressaten, Positionierung und Einschränkung

Mit der vorliegenden Arbeit sprechen wir in erster Linie *Prozessingenieure*, also Entwickler von Vorgehensmodellen, an. Wir entwickeln mit der vorliegenden Arbeit Strukturen und Mechanismen zur *projektspezifischen Anpassung* von *existierenden* Vorgehensmodellen. Die konkrete Ausformulierung der Inhalte basierend auf den entwickelten Strukturen wird zwar in den Fallstudien (vergleiche Kapitel 7) exemplarisch durchgeführt und durch die Methode in Kapitel 6 und den Kriterienkatalog in Anhang A unterstützt, steht aber nicht im Fokus dieser Arbeit.

Aus Sicht des Vorgehensmodell-*Entwicklers* bieten die in der vorliegenden Arbeit entwickelten Strukturen und Mechanismen ein Werkzeug zur Beschreibung der Projektcharakteristika, der variablen Anteile des Vorgehensmodells und der Verknüpfung zwischen beiden. Die Abstützung der Mechanismen auf bekannte Techniken der Feature-Modellierung hilft dem Prozessingenieur bei der Überprüfung der Konsistenz und bei der Analyse möglicher Konfigurationen.

Mit „variablen Anteilen“ meinen wir hier ausdrücklich nur die der projektspezifischen Anpassung unterworfenen Inhalte eines Vorgehensmodells. Ein Vorgehensmodell unterliegt in seinem Lebenszyklus üblicherweise einer Reihe von Anpassungsschritten. Anerkannt ist folgende Einteilung der Anpassungsschritte (vergleiche Abschnitt 2.2.2 und [KTF09, Kuh07]):

Tailoringstufe 1 Initiale Erstellung des Vorgehensmodells oder Anpassung eines generischen Vorgehensmodells an die Organisation.

Tailoringstufe 2 Projektspezifische Anpassung des (organisationsspezifischen) Vorgehensmodells für ein konkretes Projekt. Während sich ein einmal für

die Organisation vereinbartes Vorgehensmodell in der Regel nur langsam und punktuell ändert, wird die projektspezifische Anpassung für jedes Projekt neu durchgeführt und die resultierende Ausprägung des Vorgehensmodells ist nur für das spezielle Projekt gültig.

Ausgestaltung Die tatsächliche Projektdurchführung wird die Spielräume zwischen den durch das Vorgehensmodell beschriebenen Leitplanken ausnutzen und gegebenenfalls von den Vorgaben abweichen (müssen). Die Differenz zwischen dem idealisierten Vorgehensmodell und der gelebten Projektrealität kann ebenfalls als Anpassungsstufe des Vorgehensmodells betrachtet werden. Es wird interpretiert und beispielsweise durch nicht beschriebene Methoden ausgestaltet.

Wir verstehen unsere Arbeit als Beitrag zur *Tailoringstufe 2*. Der *Tailoringstufe 1*, also der Architektur eines an organisationsspezifische Gegebenheiten anpassbaren Vorgehensmodells bzw. einer Vorgehensmodell-Familie haben sich beispielsweise [Kuh07, Ter10] gewidmet.

Aus Sicht des Vorgehensmodell-Anwenders erlaubt das feature-basierte Tailoring die flexible Anpassung an die Bedürfnisse des Projekts. Wie das Tailoring konkret durchzuführen ist, hängt nicht zuletzt von dem durch den Prozessingenieur definierten Konfigurationsmodell und den darin beschriebenen Alternativen ab. Die Methodenbeschreibung für den Anwender in Abschnitt 6.4 gibt einige Hinweise für die Durchführung des Tailorings aus Sicht den Vorgehensmodell-Anwenders.

1.6.1 Anwendungsbereiche von Vorgehensmodellen

Bei jeder Art von Software- und Systementwicklung ist ein strukturiertes Vorgehen sinnvoll und wichtig. Allerdings unterscheidet sich die Anwendung eines Vorgehensmodells stark bei produktorientierter und projekt- bzw. auftragsorientierter Entwicklung.

Projektgeschäft

Das Projektgeschäft ist gekennzeichnet durch:

- Für jedes Projekt neu zu bestimmende Projektparameter, wie zum Beispiel Entwicklungstechnologie, Teamgröße, Sicherheitskritikalität des zu entwickelnden Systems, Verfügbarkeit von Endanwendern, usw.
- Oftmals instabile Anforderungen.
- Notwendigkeit der Einarbeitung in die Entwicklungsdomäne.
- Abhängigkeit von vertraglichen Rahmenbedingungen.

Aufgrund der zeitlichen Begrenztheit, der Einmaligkeit und den jeweils individuellen Umständen von Software- und Systementwicklungsprojekten² ist bei dieser Art der Softwareentwicklung jedes Mal individuell eine Entscheidung für eine geeignete Vorgehensweise zu treffen. Diese muss an die projektspezifischen Gegebenheiten angepasst werden. Für diesen Schritt leisten wir mit der vorliegenden Arbeit einen Beitrag.

² Auf den Projektbegriff gehen wir in Abschnitt 2.2.2 näher ein.

Produktgeschäft

Das Produktgeschäft ist gekennzeichnet durch:

- Die Rolle des Kunden wird oft von internen Rollen – beispielsweise dem Produktmanagement, Marketing, Applikationsingenieuren oder bei kleineren Unternehmungen von der Geschäftsführung – ausgefüllt. Diese wiederum ermitteln und priorisieren (in Zusammenarbeit mit den Endkunden und anderen Interessensgruppen) die Anforderungen an das zu entwickelnde System.
- Vertrautheit der Teammitglieder mit der Entwicklungsdomäne.
- In der Regel vergleichsweise stabile Anforderungen.
- Bekannte und stabile Projektparameter.

Die Entwicklung bei dieser Art der Softwareentwicklung verläuft typischerweise in Zyklen, die sich an den Release-Zyklen des Produkts orientieren. Zwischen zwei Versionen des Produkts wird der durch das Vorgehensmodell beschriebene Prozess (mindestens) einmal komplett durchlaufen. Vorlage für einen Prozess kann beispielsweise ein einfacher Wasserfall oder ein Spiralmodell sein. Zwar kann man die Entwicklung zwischen zwei Produktversionen jeweils als eigenständiges Projekt auffassen – allerdings werden sich die Umstände zwischen zwei derartigen „Projekten“ nur graduell ändern (beispielsweise aufgrund von Erkenntnissen aus interner Reflexion, kontinuierlicher Prozessverbesserung oder aufgrund von Rückmeldung der Kunden). Tailoring – also eine komplette Neuanpassung des Vorgehensmodells – ist in der Regel nicht notwendig.

In der vorliegenden Arbeit gehen wir daher von der Anwendung eines Vorgehensmodells im Projektgeschäft mit stark variierenden Projektparametern aus.

1.7 Aufbau dieser Arbeit

Die vorliegende Arbeit ist wie folgt aufgebaut:

Kapitel 2 gibt einen Überblick über den aktuellen Stand der Forschung und Praxis in Bezug auf die projektspezifische Anpassung von Vorgehensmodellen. Neben einer Begriffsbildung für Vorgehensmodell, Methode und Tailoring geben wir einen Überblick über Arbeiten zur Einordnung von Vorgehensmodellen und zu Arbeiten, welche sich mit Kriterien für die Einstufung und für die projektspezifische Anpassung befassen. Da sich das feature-basierte Tailoring wesentlich auf die Techniken der Feature Oriented Domain Analysis abstützt, befassen wir uns in Kapitel 2 weiterhin mit existierenden Arbeiten zur Prozesslinienmodellierung und geben einen knappen Überblick über die wesentlichen Konzepte der Feature-Modellierung.

Kapitel 3 analysiert die Anpassungsfähigkeit existierender Vorgehensmodelle beim projektspezifischen Tailoring. Konkret untersuchen wir die Vertreter V-Modell XT, das Eclipse Process Framework (zusammen mit seinem Metamodell SPEM), die schweizerische Projektführungsmethode HERMES und das SE Book der T-Systems International GmbH. Weiterhin analysieren wir in Kapitel 3 die Anpassungsfähigkeit leichtgewichtiger Methoden anhand der Vertreter Extreme Programming, Crystal Clear, Scrum und Feature Driven Development. Die Analyse der Anpassungsfähigkeit stellt

eine wesentliche Motivation für den in der vorliegenden Arbeit entwickelten Ansatz dar.

Kapitel 4 Der Ansatz des feature-basierten Tailorings definiert ein Metamodell zur Beschreibung der Projektcharakteristika und der Variabilitätspunkte in einem Vorgehensmodell. Er gründet deshalb auf der Annahme, dass das Vorgehensmodell selbst auf einem Metamodell basiert, welches zur Darstellung der Variabilität um die hier entwickelten (Meta-)Modelle erweitert oder mit diesen integriert werden kann. In Kapitel 4 zeigen wir anhand ausgewählter Vorgehensmodelle, wann diese Annahme berechtigt ist und für welche Art von Vorgehensmodellen diese Annahme nicht getroffen werden kann. Hierzu untersuchen wir die innere Struktur des V-Modell XT, des Metamodells SPEM und analysieren, ob und inwieweit sich agile Vorgehensweisen durch ein Metamodell unterlegen lassen.

Kapitel 5 beschreibt die theoretischen Grundlagen des feature-basierten Tailorings. Wir entwickeln ein Metamodell zur Beschreibung der Projektcharakteristika und zur Verknüpfung der Projektcharakteristika mit den Inhaltsträgern des Vorgehensmodells über definierte Variabilitätspunkte. Die Semantik der Feature-Modelle machen wir uns zunutze, um Abhängigkeiten zwischen Variabilitätspunkten zu beschreiben und skizzieren in diesem Kapitel Verfahren zur Sicherstellung einer strukturell und inhaltlich sinnvollen projektspezifischen Ausprägung des Vorgehensmodells.

Kapitel 6 In Kapitel 6 beschreiben wir die Methode zur Umsetzung und Anwendung des feature-basierten Tailorings. Im ersten Teil des Kapitels beschreiben wir eine Methode zur Umsetzung des feature-basierten Tailorings für den Prozessingenieur. Diese besteht im Kern in der Vereinbarung geeigneter Variabilitätspunkte und in der Wahl der Menge der Projektcharakteristika. Im zweiten Teil des Kapitels gehen wir auf die Durchführung des feature-basierten Tailorings aus Sicht des Vorgehensmodell-Anwenders ein.

Kapitel 7 beinhaltet zwei Fallstudien, die wir zur Evaluation der Ergebnisse dieser Arbeit durchgeführt haben. Die erste Fallstudie befasst sich mit der Anwendung des feature-basierten Tailorings auf das SE Book der T-Systems International GmbH. Die zweite Fallstudie ersetzt die Mechanismen des Tailorings im V-Modell XT durch die Konzepte des feature-basierten Tailorings. Im dritten Teil des Kapitels beschreiben wir ein kontrolliertes Experiment, bei welchem die Tailoring-Ansätze des SE Book (mit feature-basiertem Tailoring, also den Ergebnissen der ersten Fallstudie), des V-Modell XT und von EPF aus Sicht von Vorgehensmodell-Anwendern miteinander verglichen werden.

Kapitel 8 fasst die Ergebnisse dieser Arbeit zusammen, bewertet die Ergebnisse und zeigt auf, welche offenen Punkte in zukünftigen Arbeiten adressiert werden sollten.

Der Anhang A beinhaltet einen Katalog von Kriterien zur Charakterisierung von Projekten vor dem Hintergrund der projektspezifischen Anpassung eines Vorgehensmodells. Die im Katalog enthalten Kriterien wurden aus der Literatur und aus bekannten Vorgehensmodellen zusammengetragen und können als Referenz bei der Umsetzung des feature-basierten Tailorings in einem eigenen Vorgehensmodell dienen.

Ausgangslage und verwandte Arbeiten

In diesem Kapitel legen wir die Grundlagen für die Entwicklung des feature-basierten Tailorings. Wir definieren das in dieser Arbeit verwendete Verständnis eines Vorgehensmodells und den Begriff des Tailorings. Da die projektspezifische Anpassung auch als „Wahl“ eines Vorgehensmodells aufgefasst werden kann, geben wir einen Überblick über Arbeiten zur Klassifizierung und Eignungsfeststellung von Vorgehensmodellen. Weiterhin geben wir einen Überblick über existierende Arbeiten zur Anpassung von Vorgehensmodellen über den gesamten Lebenszyklus eines Vorgehensmodells. Ein Schwerpunkt der Betrachtung liegt hierbei auf solchen Verfahren, die Techniken der Produktlinienmodellierung auf Vorgehensmodelle anwenden. Den Abschluss des Kapitels bildet ein Überblick über die Theorie der Feature Oriented Domain Analysis.

Am Ende dieses Kapitels wird der Leser einen Überblick über den Stand der Forschung und Praxis in Bezug auf die vorliegende Arbeit haben. Der Leser wird Kenntnis der Ansätze zur Einordnung und Feststellung der Eignung von Vorgehensmodellen und einen Überblick über Verfahren zu deren Anpassung haben. Weiterhin wird der Leser mit den Grundlagen der Feature Oriented Domain Analysis vertraut sein. Diese bilden eine wesentliche Grundlage für den weiteren Verlauf dieser Arbeit.

Übersicht

2.1	Überblick über verwandte Arbeiten	16
2.2	Begriffe	17
2.3	Einordnung von Vorgehensmodellen	27
2.4	Charakterisierung von Projekten	33
2.5	Verfahren zur Anpassung von Vorgehensmodellen und Methoden	51
2.6	Feature-Oriented Domain Analysis	57
2.7	Zusammenfassung	63

2.1 Überblick über verwandte Arbeiten

Gegenstand dieses Kapitels ist ein Überblick über relevante Beiträge aus Forschung und Praxis. Abbildung 2.1 illustriert den Aufbau dieses Kapitels.

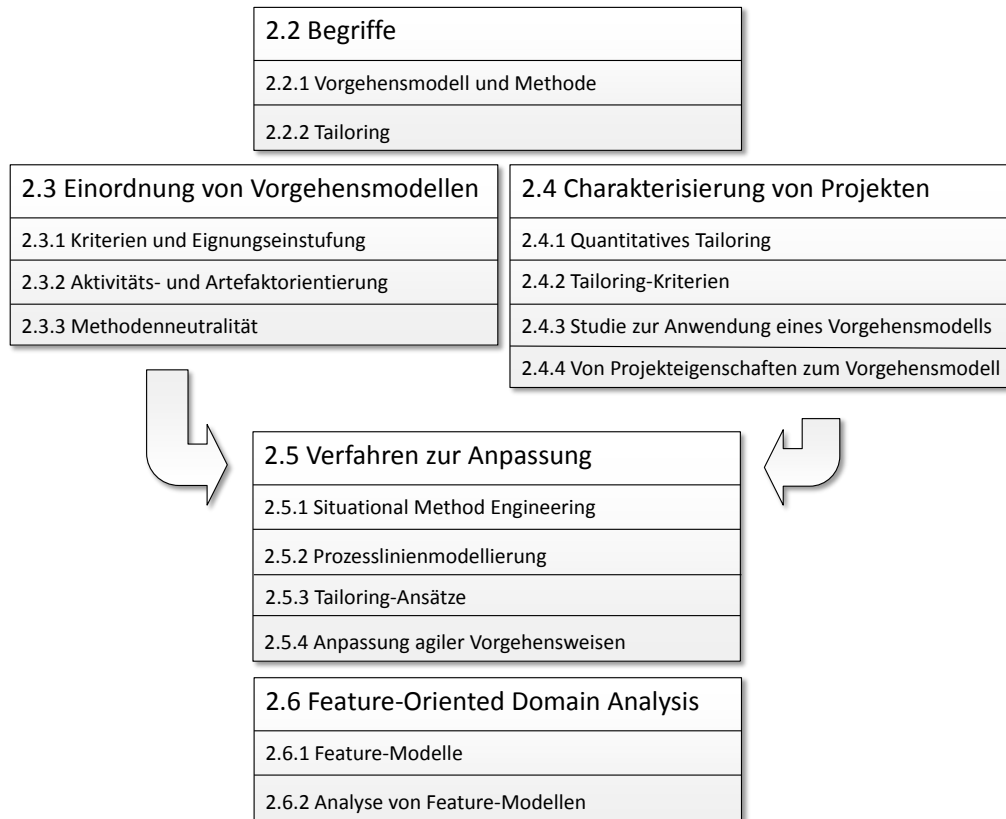


Abbildung 2.1: Struktur des Kapitels 2

Da die Begriffe Vorgehensmodell, Methode und (Entwicklungs-)Prozess zum Teil sehr unterschiedlich und zum Teil synonym verwendet werden, schaffen wir im ersten Abschnitt dieses Kapitels ein gemeinsames Verständnis für die in der vorliegenden Arbeit verwendeten Begriffe. Insbesondere die Begriffe *Vorgehensmodell* und *Methode* verwenden wir in der vorliegenden Arbeit mit einer klar voneinander abgegrenzten Bedeutung. Im zweiten Teil des Abschnitts 2.2 grenzen wir ein, was wir in dieser Arbeit unter *Tailoring* eines Vorgehensmodells verstehen. Auch dieser Begriff wird in der Literatur sehr unterschiedlich und für verschiedene Tätigkeiten im Anpassungsprozess eines Vorgehensmodells verwendet. Bei der Begriffsbildung zu den Anpassungsschritten orientieren wir uns an der Terminologie des V-Modell XT.

Vor der Anpassung eines Vorgehensmodells an die organisations- und projektspezifischen Gegebenheiten steht immer die Wahl oder Vereinbarung eines Vorgehensmodells als solches. Im Abschnitt 2.3 betrachten wir daher die Literatur zur Wahl von Vorgehensmodellen. Kriterienkataloge und quantitative Ansätze zur Wahl eines Vorgehensmodells oder zu dessen retrospektiver Beurteilung – beispielsweise im Rahmen kontinuierlicher Prozessverbesserung – geben nicht zuletzt Hinweise und Aufschluss über Bereiche eines Vorgehensmodells, die einer projektspezifischen Anpassung unterworfen werden können oder sollten.

In Abschnitt 2.4 setzen wir uns mit der Charakterisierung von Projekten vor

dem Hintergrund der projektspezifischen Anpassung von Vorgehensmodellen auseinander. Welche Charakteristika nehmen Einfluss auf die projektspezifische Ausprägung eines Vorgehensmodells und in welcher Weise wirken sie?

Im Abschnitt 2.4.3 stellen wir eine in Zusammenarbeit mit Partnern der T-Systems International GmbH und der Telekom Laboratories durchgeführte Studie zum Einsatz und Anpassungsbedarf des Vorgehensmodells der T-Systems International GmbH dar. Die Ergebnisse dieser für *ein* Vorgehensmodell in *einem* Unternehmen durchgeführte Studie betten wir ein in den Kontext vergleichbarer Studien.

In Abschnitt 2.5 betrachten wir allgemein existierende Ansätze zur Anpassung von Vorgehensmodellen. In der Literatur und in der aktuellen Forschung und Praxis stellen wir dabei zwei Strömungen fest: Anpassung als *Konfiguration* eines als Obermenge möglicher Projekte definierten Vorgehensmodells und Anpassung als *Konstruktion* eines Vorgehensmodells durch Zusammensetzung geeigneter Bausteine. Die erste Herangehensweise entspricht dem *Tailoring* im eigentlichen Sinne, die zweite wird üblicherweise als (*Situational*) *Method Engineering* bezeichnet.

Da wir mit dem feature-basierten Tailoring einen Ansatz zur Anpassung von Vorgehensmodellen basierend auf Feature-Modellen vorschlagen, legen wir im zweiten Teil dieses Abschnitts besonderes Augenmerk auf existierende Bestrebungen, Techniken der Produktlinienmodellierung bei der Anpassung von Vorgehensmodellen zu nutzen.

Im Abschnitt 2.6 geben wir einen Überblick über die Techniken der *Feature-Oriented Domain Analysis*. Diese bilden eine Grundlage für die in Kapitel 5 entwickelten Konzepte.

2.2 Begriffe

Im folgenden Abschnitt definieren wir wesentliche Begriffe. Besonders zu klären sind im Zusammenhang mit der vorliegenden Arbeit die Unterschiede bzw. Überschneidungen der Begriffe *Vorgehensmodell* und *Entwicklungs-Methode*, die Unterscheidung zwischen artefakt- und aktivitätsorientierten Vorgehensweisen und der Begriff des *Tailorings*.

2.2.1 Vorgehensmodell und Methode

Vorgehensmodelle und Entwicklungsmethoden dienen dazu, die Erfolgswahrscheinlichkeit von Softwareentwicklungsprojekten zu steigern, indem sie bewährte Praktiken, Abläufe und Organisationsformen der Softwareentwicklung zusammenfassen und wiederholbar machen.

Für den Einsatz von Vorgehensmodellen werden in [Gna05] folgende entscheidende Beweggründe genannt:

- Ein Vorgehensmodell soll durch die Definition einheitlicher Begriffe und Ergebnisse zur Verbesserung der Kommunikation der Projektbeteiligten beitragen.
- Ein standardisiertes Vorgehen soll die Vollständigkeit und Qualität der zu liefernden Ergebnisse gewährleisten.

2.2 Begriffe

- Ein Vorgehensmodell zielt in mannigfaltiger Weise auf eine Eindämmung der Entwicklungskosten beziehungsweise der Kosten über den gesamten Lebenszyklus eines Systems ab.
- Ein einheitlich verwendetes Vorgehensmodell stellt die Nachvollziehbarkeit und Wiederholbarkeit des Vorgehens und Prozesses sicher.

Insbesondere im Hinblick auf (agile) Vorgehensweisen sind folgende Punkte zu ergänzen:

- Ein Vorgehensmodell oder eine Methode soll eine *Kultur* im Team oder im Unternehmen verankern. Insbesondere in agilen Ansätzen finden sich häufig Aussagen über das menschliche Miteinander im Team: offener und toleranter Umgang mit Fehlern des Einzelnen, *Common Ownership*, etc.
- Ein Vorgehensmodell soll Vertrauen schaffen. In das Innere eines Projekts werden von einem Vorgehensmodell oder von der *Kultur* einer Methode stabilisierende Effekte erwartet.

Mit Blick auf den Kunden oder Auftraggeber kann eine etablierte Vorgehensweise oft Voraussetzung für die Durchführung des Projekts sein (wenn beispielsweise bestimmte Zertifizierungsniveaus gefordert sind). Auch die häufige und regelmäßige Einbeziehung des Kunden in die Entwicklung (wie typischerweise von agilen Vorgehensweisen gefordert) kann vertrauensstiftend wirken.

Die Begriffe *Vorgehensmodell* oder *Entwicklungsprozess* und *Entwicklungsmethode* werden häufig synonym verwendet. Um Missverständnisse zu vermeiden, nehmen wir für den weiteren Verlauf der Arbeit eine Abgrenzung der Begriffe vor. Erstens ist ein gewisses Maß der eingangs beschriebenen Diskrepanz zwischen reichhaltigen Vorgehensmodellen und leichtgewichtigen Methoden einer Verwechslung bzw. Vermischung der Begriffe zuzuschreiben und zweitens beschäftigen wir uns in der vorliegenden Arbeit in erster Linie mit der Anpassung von *Vorgehens-Modellen*.

Vorgehensmodell

Ein Vorgehensmodell für die Softwareentwicklung (oft auch Softwareentwicklungsprozess genannt) abstrahiert von konkreten Softwareentwicklungsprojekten und beschreibt Abläufe und Strukturen in einer idealisierten Form. Damit liegen wir auf einer Linie mit [Gna05], dessen Begriff des Vorgehensmodells wir weitgehend übernehmen:

Definition 1 (Vorgehensmodell):

Ein **Vorgehensmodell** legt einen standardisierten organisatorischen Rahmen für den Ablauf eines Entwicklungsprojektes in Form einer Beschreibung der zu erstellenden Produkte, der durchzuführenden Aktivitäten und der Rollen fest. Ein Vorgehensmodell ist damit ein Modell beziehungsweise eine Abstraktion der Vorgehensweise im konkreten Projekt. Ein Vorgehensmodell kann auch als „Handlungsmuster“ bezeichnet werden. Als synonym betrachten wir den ebenfalls gebräuchlichen Begriff Prozessmodell.

In der vorliegenden Arbeit fordern wir von einem Vorgehensmodell, dass sich zumindest die in Abbildung 2.2 dargestellten Teilaspekte wiederfinden:

Artefaktmodell Das Artefaktmodell beschreibt, *was* im Projekt zu tun ist, also die im Laufe eines Projekts zu erstellenden (Zwischen-)Ergebnisse. Dies

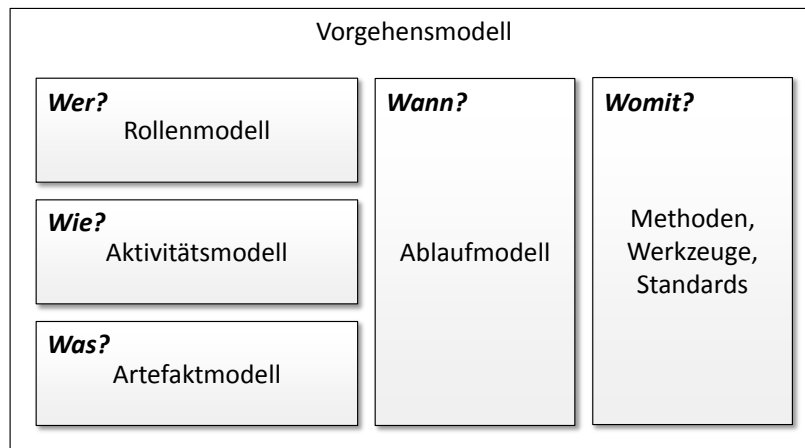


Abbildung 2.2: Submodelle eines Vorgehensmodells (angelehnt an [Kuh07])

umfasst sowohl das tatsächliche Projektergebnis – beispielsweise die Lieferung an den Kunden – als auch interne Artefakte wie Test- oder Prüfereports.

Aktivitätsmodell Das Aktivitätsmodell beschreibt, *wie* mit den Artefakten umgegangen wird. Dies umfasst die Erstellung, Bearbeitung und Verwendung von Artefakten. Im Aktivitätsmodell werden unter anderem auch konkrete Modellierungs-*Methoden* beschrieben, etwa zur Entwicklung einer Systemarchitektur.

Rollenmodell Im Rollenmodell wird vereinbart, *wer* welche Tätigkeiten und Verantwortungen hat. Rollen sind oft in der Form eines Fähigkeitsprofils beschrieben: welche Kenntnisse sollte ein Projektbeteiligter mitbringen, um eine Rolle auszufüllen?

Ablaufmodell Der Ablauf ordnet die Erstellung von Artefakten und die dazugehörigen Aktivitäten zeitlich an. Dies erfolgt durch die Definition von Meilensteinen oder Phasen. Das Ablaufmodell legt damit fest, *wann* Tätigkeiten durchgeführt werden und Ergebnisse vorliegen.

Werkzeuge und Methoden Ein Vorgehensmodell beinhaltet üblicherweise Hilfsmittel, die die Projektbeteiligten bei der Anwendung des Vorgehens unterstützen. Werkzeuge und Methoden im Vorgehensmodell beschreiben, *womit* das Vorgehensmodell selbst aber auch die darin beschriebenen Aktivitäten bearbeitet werden.

Die Schwerpunktsetzung und der Detaillierungsgrad der einzelnen Bestandteile können sich von Vorgehensmodell zu Vorgehensmodell natürlich stark unterscheiden.

In [FK07] als „Universelle Vorgehensmodelle“ bezeichnete allgemeine Modelle eines Projektablaufs betrachten wir in dieser Arbeit nicht weiter. In diese Kategorie fallen Modelle wie zum Beispiel:

- das Wasserfallmodell (dessen erste Nennung [Roy70] zugeschrieben wird und von Barry W. Boehm seinen Namen bekommen haben soll).
- das Spiralmodell [Boe88].
- prototypische Entwicklung [Bal00].
- inkrementelle Entwicklung [Som01].

2.2 Begriffe

Derartige Modelle können das Fundament für die Ausgestaltung eines kompletten Vorgehensmodells im obigen Sinne sein. Sie erfüllen allerdings nicht alle unsere Forderungen, die wir an ein *komplettes* Vorgehensmodell stellen. Teile des Artefaktmodells und/oder des Aktivitätsmodells werden in diesen Modellen zwar angedeutet, bleiben aber stets so vage, allgemein und unverbindlich, dass sich im Rahmen dieser Arbeit keine Aussagen über die Anpassung solcher Modelle machen ließen.

Metamodellierung

Im weiteren Verlauf dieser Arbeit ist öfter die Rede von (Vorgehens-)Modellen, Metamodellen und Meta-Metamodellen. Im folgenden Abschnitt klären wir knapp die Bedeutung dieser Begriffe.

Ein Vorgehensmodell zur Software- und Systementwicklung ist ein *Modell* der Wirklichkeit in einem Softwareentwicklungsprojekt. Ein Modell ist durch folgende Eigenschaften charakterisiert:

- Einem Modell liegt ein konkreter Sachverhalt oder Gegenstand zugrunde. Das Modell *beschreibt* diesen Sachverhalt oder Gegenstand [Str98].
- Es *abstrahiert* von den Details der Wirklichkeit (bzw. des modellierten Gegenstands) und ermöglicht dadurch die Auseinandersetzung mit dem Gegenstand (in unserem Fall einem Softwareentwicklungsprojekt) auf einem höheren Abstraktionsniveau. Davon erwartet man sich erhöhte Produktivität und erleichterten Erkenntnisgewinn [BFH⁺10].
- Es *vereinfacht* den Betrachtungsgegenstand, indem unwesentliche oder nebensächliche Sachverhalte der Wirklichkeit (bzw. des modellierten Gegenstands) außer Acht gelassen werden. Ein Modell reduziert damit die zu beherrschende Komplexität.

Ist das Modell seinerseits Gegenstand der Modellierung, so spricht man von *Metamodellierung* [Str98]. Ein *Metamodell* definiert die Sprache zur Beschreibung des Modells. Das Modell stellt damit eine *Instanz* des Metamodells dar. Für die Betrachtung der verschiedenen Modellebenen ist die in Abbildung 2.3 dargestellte Einteilung in die Ebenen M0 bis M3 der Object Management Group (OMG) weit verbreitet und akzeptiert.

Modellierungs-					
Ebene	ebene nach MOF	UML	SPEM	V-Modell XT	Crystal Clear
M3	Meta-Metamodell	MOF	MOF		
				V-Modell XT	
M2	Metamodell	UML-Sprache	SPEM 2.0	Metamodell	
M1	Modell	Klassenmodell	Method Library	V-Modell XT	Crystal Clear
		Laufzeit-	Projektmodell/	Projektmodell/	Projektmodell/
M0	Instanzen	/Objektmodell	Projektplan	Projektplan	Projektplan

Abbildung 2.3: Modellierungsebenen nach MOF (angelehnt an [Kuh07] und [OMG08])

Die Ebenen stellen von unten (M0) nach oben (M3) eine Abstraktion des Betrachtungsgegenstandes dar:

- M0** Das Modell auf dieser Ebene beschreibt ein *Laufzeit-* oder *Objektmodell*. In Bezug auf Softwareentwicklungsprojekte handelt es sich hierbei um die konkrete *Projektinstanz* mit Rollenzuweisungen, einer Systemzerlegung, einem Projektplan usw.

- M1** Abstrahiert vom Betrachtungsgegenstand auf der Ebene M0. In einer Analogie zur Objektorientierung befänden sich auf M0 die Objekte und auf M1 die Klassen. In Bezug auf Softwareentwicklungsprojekte befinden sich auf M1 die eigentlichen *Vorgehensmodelle* und *Methoden*.
- M2** Auf dieser Ebene befindet sich das „Modell des Modells“ – das Metamodell. Das Metamodell definiert die Sprache zur Beschreibung des auf M1 befindlichen Modells.
- M3** In der Ebene M3 sind Meta-Metamodelle anzusiedeln. Diese definieren die Sprache zur Beschreibung des Metamodells. Da man die „Metaisierung“ theoretisch beliebig weiter treiben könnte, wird ein Modell auf Ebene M3 üblicherweise mit Mitteln der durch das Modell selbst definierten Sprache beschrieben.

Wie in Abbildung 2.3 skizziert, lassen sich sowohl die UML 2.0 als auch das SPEM 2.0 Metamodell mit Hilfe der Meta Object Facility (MOF) [OMG11a] beschreiben. Auf das Metamodell SPEM gehen wir in Abschnitt 3.2.2 detaillierter ein. Das V-Modell XT Metamodell ließe sich höchstwahrscheinlich ebenfalls mit Ausdrucksmitteln der MOF beschreiben. Derartige Arbeiten sind nicht bekannt, allerdings werden in [Fri06] zumindest die Möglichkeiten der Überführung von Inhalten des V-Modell XT auf die Strukturen von EPF analysiert. Mit dem V-Modell XT und seinem Metamodell beschäftigen wir uns in Abschnitt 3.2.1 eingehend. In Abbildung 2.3 ist als beispielhafter Vertreter für eine leichtgewichtige Vorgehensweise *Crystal Clear* [Coc04] dargestellt. Diese Vorgehensweise (vergleiche auch den folgenden Abschnitt 2.2.1) kann zwar als ein *Modell* eines konkreten Projekts auf Ebene M0 aufgefasst werden – sie ist aber nicht durch ein formales Metamodell unterlegt.

Mit diesen Begriffen ließe sich die Frage der vorliegenden Dissertation so formulieren: Wie müssen die Strukturen auf M2 aussehen, damit ein auf Ebene M1 umfassend und allgemein beschriebenes Vorgehensmodell mittels *Tailoring* so angepasst werden kann, dass dessen Anwendbarkeit auf Ebene M0 gesteigert wird.

Methode

Die Begriffe Vorgehensmodell und (Entwicklungs-)Methode werden häufig synonym verwendet. Die folgenden, häufig zitierten Definitionen von [Bri96] offenbaren allerdings feine, aber wesentliche Unterschiede:

A **method** is an approach to perform a systems development project, based on a specific way of thinking, consisting of directions and rules, structured in a systematic way in development activities with corresponding development products.

Methode nach [Bri96]

The **methodology of information system development** is the systematic description, explanation and evaluation of all aspects of methodical information systems development.

Systementwicklungsmethode nach [Bri96]

Method Engineering is the engineering discipline to design, construct and adapt methods, techniques and tools for the development of information systems.

Method Engineering nach [Bri96]

2.2 Begriffe

Eine weitere, etwas spezifischere Definition für Softwareentwicklungsmethoden findet sich bei [Som01]:

Software Engineering Methods are structured approaches to software development which include system models, notations, rules, design advice and process guidance.

Software Engineering Methode nach [Som01]

Bei einer Methode nach diesen Definitionen ist die Rede von einem „Ansatz zur Durchführung“, einer „speziellen Art, zu denken“, von „Notationen“ und von „Hinweisen und Regeln“. Keine Rede ist von Arbeitsergebnissen, Aktivitäten und Verantwortlichkeiten und Beziehungen zwischen diesen Einheiten.

Wenn ein Vorgehensmodell, wie [Gna05] schreibt, einem Kochrezept gleicht, so entsprechen (insbesondere agile) Methoden eher Hinweisen der Art „Verwenden Sie zum Kochen der Pasta einen geeigneten Topf“, „Kochen Sie nicht alle Gänge auf einmal“, „Schmecken Sie Ihre Gerichte regelmäßig ab“, „Kochen Sie lieber nicht alleine“ oder „Jedes Gericht ist gut, so lange es Ihren Gästen schmeckt.“

In der Literatur zu Vorgehensmodellen wird der Begriff Methode häufig verwendet für Handlungsanweisungen, Beschreibungstechniken oder Werkzeuge zur Durchführung einer Aktivität.

Im V-Modell XT finden sich beispielsweise *Methodenreferenzen* zur Datenbankmodellierung mittels ER-Diagrammen, zur Modellierung funktionaler Anforderungen mittels UML oder zur Meilensteintrendanalyse im Projektmanagement. Wir verwenden den Begriff *Methode* für den weiteren Verlauf dieser Arbeit in diesem Sinne.

Fazit

Wie eingangs erwähnt, ist der Widerspruch zwischen reichhaltig definierten Vorgehensmodellen und (agilen) Methoden nur ein scheinbarer. Vorgehensmodelle machen meist keine Aussagen zum menschlichen Umgang und Miteinander im Team, zu den Anforderungen an den Einzelnen, zu bewährten Praktiken und konkreten Beschreibungstechniken. (Agile) Methodenbeschreibungen wiederum beschränken sich in der Betrachtung zumeist auf ein einzelnes Entwicklungsteam. Die Schnittstelle zum Kunden, zu Teams an geographisch verteilten Standorten und zu anderen Organisationseinheiten bleiben außerhalb der Betrachtung. Zudem werden Arbeitsergebnisse und Beziehungen zwischen diesen meist nur informell beschrieben.

Im Hinblick auf die Anpassung von Vorgehensweisen geht es einerseits um die sinnvolle Integration von Vorgehensmodellen und Methoden und andererseits um die Steigerung der „Passgenauigkeit“ von Vorgehensmodellen, insbesondere der Reduktion unnötigen bürokratischen Mehraufwandes. Mit dem Anpassungsbegriff setzen wir uns im folgenden Abschnitt auseinander.

2.2.2 Tailoring

Zur Annäherung an den Begriff des Tailorings betrachten wir zunächst den Anwendungskontext eines Vorgehensmodells. Wie in Abschnitt 1.6.1 ausgeführt, betrachten wir in der vorliegenden Arbeit überwiegend Vorgehensmodelle bei ihrer Anwendung im Projektgeschäft.

Ein Projekt wird in der DIN 69901 [Deu87] wie folgt definiert:

Definition 2 (Projekt):

Ein **Projekt** ist ein Vorhaben, das im Wesentlichen durch Einmaligkeit der Bedingungen in ihrer Gesamtheit gekennzeichnet ist, wie z. B.: Zielvorgabe, zeitliche, finanzielle, personelle oder andere Bedingungen, Abgrenzungen gegenüber anderen Vorhaben und projektspezifische Organisation.

Wesentliches Merkmal von Softwareentwicklungs-Projekten ist also ihre Einzigartigkeit.

Reichhaltige Standardvorgehen sind bewusst so generisch gehalten, dass sie in einer großen Vielfalt von Organisationen und Projekten anwendbar sind. Aus diesem Grund sind reichhaltige Standardvorgehen – insbesondere was die konkrete Unterstützung für die Projektbeteiligten betrifft – zwangsweise unverbindlich und unspezifisch. Sie stellen eine Vorlage dar für das Vorgehen einer Teilmenge der Gesamtheit aller Projekte (vergleiche Abbildung 2.4). Diese Teilmenge entspricht den Projekten, die zum Vorgehensmodell konform durchgeführt werden (können). Beim *Tailoring* nutzen wir das Wissen über das konkrete Projekt auf Ebene M0, um das Vorgehensmodell auf Ebene M1 zu spezialisieren. Damit wird die Menge der zum resultierenden, projektspezifischen Vorgehensmodell konformen Projekte um das konkrete Projekt enger gefasst.

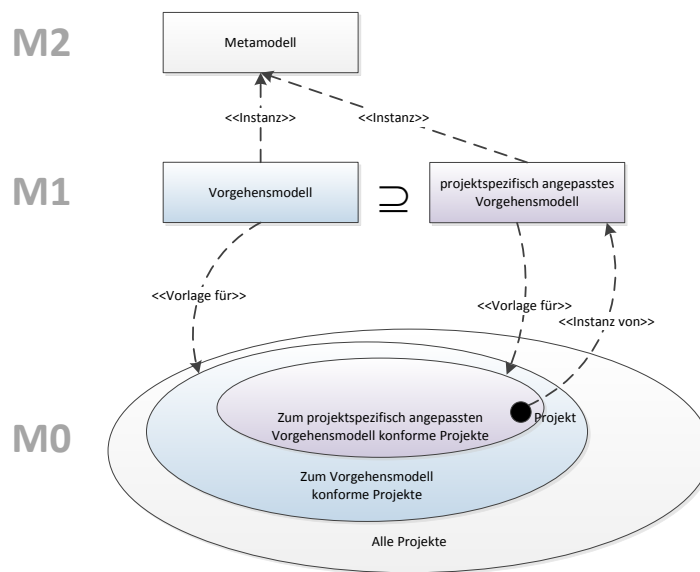


Abbildung 2.4: Was ist Tailoring?

Bei der Betrachtung der Anpassungsmechanismen ist es hilfreich, den Zeitpunkt der Anpassung im Lebenszyklus des Vorgehensmodells zu berücksichtigen. In Anlehnung an [Kuh07] durchläuft ein Vorgehensmodell die in Abbildung 2.5 dargestellten Phasen:

Erstellung In dieser Stufe wird das Vorgehensmodell initial erstellt und organisationsweit geltende Festlegungen (beispielsweise Terminologie, Artefaktmodell, etc.) getroffen.

Tailoring Auf einem in der ersten Stufe entwickelten Modell aufbauend wird in dieser Stufe das Vorgehensmodell an die konkrete Situation des Projekts angepasst. Es wird eine projektspezifische Ausprägung erstellt und

2.2 Begriffe

Inhalte, wie z. B. Vorlagen für Arbeitsergebnisse und der Projektplan, instanziiert.

Ausgestaltung Während der Projektdurchführung werden die Vereinbarungen des Vorgehensmodells mit Leben gefüllt. Dabei ist eine flexible Interpretation und praktische Ausgestaltung der darin beschriebenen Vorgaben unumgänglich. Die Projektdurchführung selbst (mit der Ausnutzung von Spielräumen) kann daher ebenfalls als Anpassungsphase des Vorgehensmodells betrachtet werden.

Reflexion Im Zuge einer kontinuierlichen Prozessverbesserung werden die während der Anwendung des Vorgehensmodells gesammelten Erfahrungen und Erkenntnisse genutzt, um das Vorgehensmodell in einer förderlichen Weise fortzuschreiben.

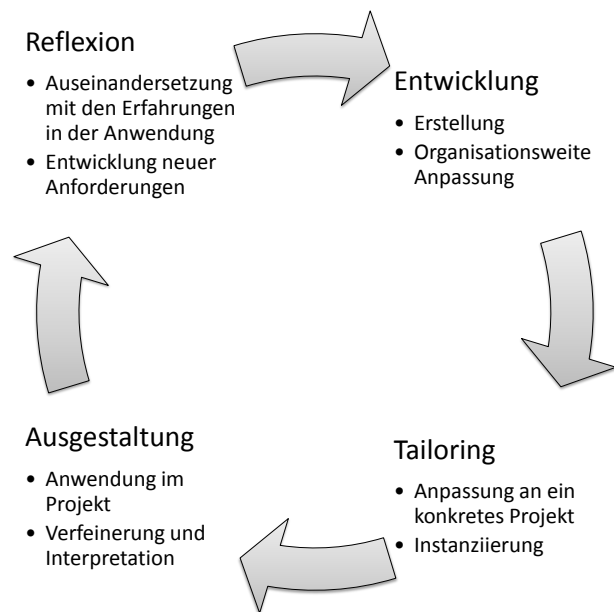


Abbildung 2.5: Lebenszyklus eines Vorgehensmodells

Auch wenn in [Kuh07] die Erstellung eines Vorgehensmodells und dessen organisationsspezifische Anpassung als *Tailoringstufe 1* und das projektspezifische Tailoring als *Tailoringstufe 2* bezeichnet werden, verwenden wir den Begriff *Tailoring* in der vorliegenden Arbeit nur im Sinne der projektspezifischen Anpassung. Die *Tailoringstufe 1* nach [Kuh07] begreifen wir als Phase der Erstellung eines Vorgehensmodells.

Folgt man der Typisierung von [Chr92], lassen sich hinsichtlich der Anpassung drei grobe Typen von Vorgehensmodellen unterscheiden:

Dach-Modell Ein derartiges Modell enthält eine Obermenge der für ein einzelnes Projekte relevanten Beschreibungen. Die Anpassung an ein konkretes Projekt erfolgt hier durch Entfernen der nicht benötigten Teile.

Kern-Modell Kern-Modelle umfassen alle solche Entwicklungsprodukte und Aktivitäten, von denen angenommen werden kann, dass sie in jedem Projekt Relevanz besitzen. Die Anpassung erfolgt durch Ergänzung und projektabhängige Ausgestaltung der Vorgaben.

Modulares Bausteinmodell Derartige Modelle beinhalten eine Bibliothek von Methoden(-bausteinen), aus denen ein Vorgehensmodell für ein konkretes

Projekt zusammengesetzt werden kann. Üblicherweise werden ebenfalls die Regeln für die korrekte Zusammensetzung der Bausteine beschrieben.

Wir verwenden den Begriff *Tailoring* in dieser Arbeit gleichbedeutend mit der Anpassungstätigkeit des ersten Typs – eines Dach-Modells. Bei den anderen beiden Anpassungstypen handelt es sich nicht um eine Anpassung im eigentlichen Sinne, als vielmehr um eine (teilweise) projektspezifische Neuerstellung des Vorgehensmodells. Dieses Verständnis ist in Übereinstimmung mit [PJg06]: hier wird die Anpassung einer Vorgehensweise an eine konkrete Situation „Methodenkonfiguration“ genannt. Dabei handele es sich um eine spezielle Form des *Situational Method Engineering* (siehe auch 2.5.1), bei welcher eine gewählte Methode als Ausgangsbasis für die Konfiguration dient. Dies stünde im Gegensatz zu den meisten Ansätzen des Method Engineering, welche davon ausgehen, dass eine situationsgerechte Methode durch das Zusammensetzen einer (üblicherweise großen) Anzahl kleiner, „atomarer“ Methodenbausteine entsteht [BSH99, Har97] (dies entspräche insbesondere dem modularen Bausteinmodell nach [Chr92]). Laut [PJg06] sei jedoch oft die bedeutendere Frage, welche Teile einer Methode gefahrlos weggelassen werden können.

Wie in [Inf04] ausgeführt, ist das Hauptziel des Tailorings, für jedes Projekt zu gewährleisten, dass der eingesetzte Aufwand den Projektzielen entspricht bzw. dienlich ist und dass, orientiert an der Aufgabenstellung, nur die sachlich erforderlichen Ergebnisse erarbeitet und somit auch nur die dazu notwendigen Tätigkeiten durchgeführt werden.

In [GQ95] wird Tailoring beschrieben als „the act of adjusting the definitions and/or of particularizing the terms of a general process description to derive a new process applicable to an alternative (and probably less general) environment“.

[Gna05] schreibt zur Anpassbarkeit: „Unter Anpassbarkeit verstehen wir die einem Vorgehensmodell inhärente Fähigkeit, ohne zu großen Aufwand an den Kontext eines Projektes anpassbar zu sein.“ Besondere Betonung verdienen hier erstens *die Fähigkeit, ohne großen Aufwand anpassbar zu sein* und zweitens die *Anpassung an den Kontext eines Projektes*.

Für die vorliegende Arbeit fassen wir den Begriff des Tailorings noch etwas enger:

Definition 3 (Tailoring):

Unter **Tailoring** verstehen wir die Anpassung eines vorgegebenen Vorgehensmodells zu Projektbeginn auf Basis der *Projektcharakteristika* durch einen Anwender des Vorgehensmodells. Durch das Tailoring wird eine für das Projekt spezialisierte Ausprägung des allgemeinen Vorgehensmodells erstellt. Dabei sind die Variationspunkte, die Regeln und die Methode zur Durchführung des Tailorings klar definiert.

Die Betonung liegt hier auf Anpassung *eines vorgegebenen* Vorgehensmodells. Im Gegensatz zur organisationsspezifischen Anpassung (siehe z. B. Abschnitt 3.2.1) und dem Situational Method Engineering (vergleiche Abschnitte 2.5.1 und 3.2.2) werden beim Tailoring keine Vorgehensmodellinhalte neu erstellt. Das Tailoring zu Projektbeginn trifft mit diesem Verständnis eine Auswahl aus vorhandenen – beispielsweise mit den Mitteln des Method Engineering erstellten – Inhalten.

Das Tailoring ist eine Tätigkeit, die in der Regel *durch einen Anwender* des Vorgehensmodells – insbesondere nicht durch einen Prozessingenieur – durchge-

2.2 Begriffe

führt wird. Vom Anwender ist in der Regel nicht zu erwarten, dass er eine weitreichende Anpassung, beispielsweise die Zusammenstellung eines kompletten Vorgehensmodells durch Kombination von Methodenbausteinen wie in EPF (siehe 3.2.2) durchführen kann (und will). Weiterhin sollte „trotz Tailoring“ eine gewisse Verbindlichkeit des Vorgehensmodells und die Erfüllung organisatorischer Vorgaben auch in projektspezifischen Ausprägungen sichergestellt sein. Wären die Variationspunkte und Operationen des Tailorings nicht klar vereinbart, würde dem Anwender beim Tailoring erstens unnötige Arbeit aufgebürdet und zweitens könnte die Konformität der resultierenden Ausprägung mit organisatorischen Vorgaben nicht sichergestellt werden.

Beim Tailoring können zwei Dimensionen unterschieden werden:

Anpassung in der Breite Unter Anpassung in der Breite verstehen wir die Reduktion des Vorgehensmodells auf die Inhalte, die für die Durchführung des Projekts relevant sind.

Ist im Projekt beispielsweise nur die Anforderungsermittlung Gegenstand, weil die technische Realisierung von einem Unterauftragnehmer durchgeführt wird, so fallen alle Vorgaben zur Entwicklung von Software weg, bzw. es bleiben nur die überwachenden Tätigkeiten übrig. Dafür sind Schnittstellenvereinbarungen zum Unterauftragnehmer und definierte Abnahmekriterien von größerer Bedeutung als bei einer Eigenentwicklung. Ebenso unterscheidet sich die Vorgehensweise bei der Entwicklung eines eingebetteten Systems mit Hard- und Softwarekomponenten vom Vorgehen bei Entwicklung einer Webanwendung.

Spezialisierung in der Tiefe Unter Spezialisierung in der Tiefe verstehen wir die Bereitstellung spezifischer, auf das Projekt zugeschnittener Hinweise, Richtlinien und Methoden.

Wird beispielsweise die Implementierungstechnologie bei der Anpassung des Vorgehensmodells berücksichtigt, lassen sich wesentlich konkretere Hinweise, z. B. in Form von Programmierrichtlinien, zur Verfügung stellen, als wenn das Vorgehensmodell auf einer allgemeinen, abstrakten Ebene bleibt.

Abbildung 2.6 veranschaulicht die Aufgabe der projektspezifischen Anpassung in der Breite und in der Tiefe: der helle Rahmen oben symbolisiert den Umfang der Vorgaben eines allgemeingültigen Vorgehensmodells. Das Vorgehensmodell beinhaltet zahlreiche Vorgaben, die für das konkrete Projekt (symbolisiert durch den grau hinterlegten Rahmen) nicht relevant sind. Gleichzeitig bietet das Vorgehensmodell wenig für die konkrete Projektdurchführung nützliche Hinweise und Hilfestellungen in Form von Methodenbeschreibungen. Die projektspezifische Anpassung eines Vorgehensmodells dient einerseits der Entfernung für das Projekt nicht relevanter Anteile (dargestellt durch die Pfeile unten in der Abbildung 2.6) und andererseits der Ausgestaltung mit vertiefenden Hinweisen und Hilfestellungen (dargestellt durch die Pfeile rechts in der Abbildung). Das Ergebnis der projektspezifischen Anpassung ist in Abbildung 2.6 durch den hellgrauen Rahmen dargestellt. Da auch eine projektspezifisch angepasste Ausprägung des Vorgehensmodells immer ein idealisiertes Abbild der Realität bleibt, ist das Ergebnis nicht deckungsgleich mit dem *tatsächlichen Vorgehensmodell* bei der Durchführung des Projekts (dem grau hinterlegten Rahmen).

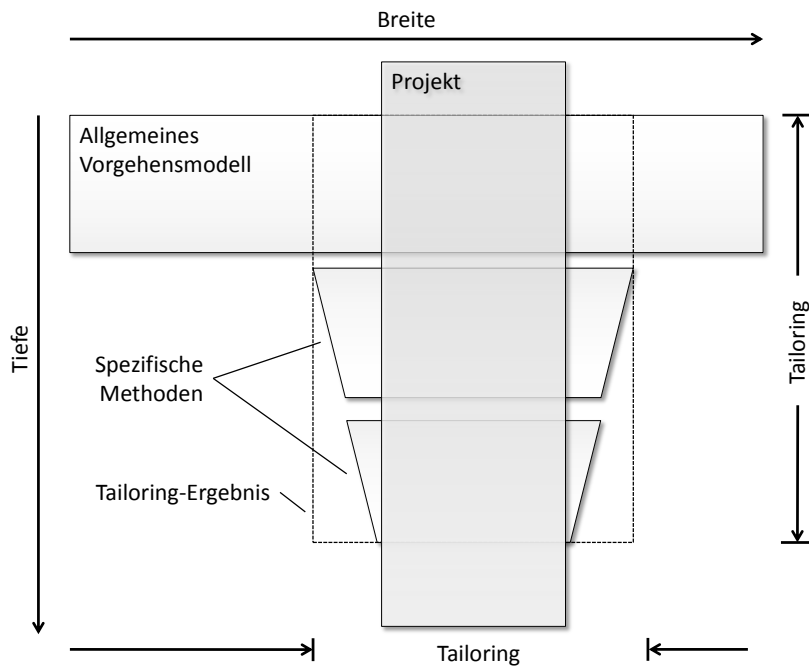


Abbildung 2.6: Tailoring in der Breite und in der Tiefe

2.3 Einordnung von Vorgehensmodellen

Ein entscheidender Schritt bei der Vereinbarung eines Vorgehensmodells für ein Projekt ist die Wahl der Ausgangsbasis. Durch die flexible Anpassungsfähigkeit des feature-basierten Tailorings lässt sich zwar die Passgenauigkeit der Vorgehensweise für die konkrete Projektsituation steigern – der „Charakter“ eines Vorgehensmodells bleibt dabei jedoch erhalten (und muss erhalten bleiben). In diesem Abschnitt befassen wir uns mit der Charakterisierung von Vorgehensmodellen und diesbezüglichen Beiträgen der Literatur.

2.3.1 Kriterien und Eignungseinstufung

In [FK07] werden etablierte Vorgehensmodelle kategorisiert und ihre Verbreitung in der deutschen Software-Industrie untersucht. Die Kategorisierung erfolgt dabei anhand folgender Kriterien:

Enge der Projektvorgaben Mit diesem Kriterium beschreiben die Autoren, wie flexibel das Vorgehensmodell für unterschiedliche Projektsituationen einsetzbar ist. Eignet es sich beispielsweise für die Durchführung eines sicherheitskritischen Systems? Ist es geeignet für kleine Projektteams? Ist das Vorgehensmodell hilfreich bei Projekten in denen die Time To Market ein kritischer Erfolgsfaktor ist?

Umfang Mit diesem Kriterium wird untersucht, inwieweit das Vorgehensmodell den Lebenszyklus eines Projekts abdeckt. Die Autoren verstehen unter diesem Kriterium einerseits die Abdeckung in der Breite (sind alle Phasen berücksichtigt?) und andererseits die methodische Unterstützung in der Tiefe (enthält das Vorgehensmodell konkrete Hinweise für die Anforderungserhebung oder zum Testen?).

Abstraktionsgrad Mit diesem Kriterium prüfen die Autoren, wie detailliert ein-

2.3 Einordnung von Vorgehensmodellen

zelne Arbeitsschritte im Vorgehensmodell beschrieben sind. Abstrakte, *universelle Modelle* beschreiben prinzipielle Schritte, während *Atomare Vorgehensmodelle* ausführliche und detaillierte Beschreibungen einzelner Aktivitäten und Arbeitsergebnisse angeben.

Erfüllung der Anforderungen Hier untersuchen die Autoren, inwieweit das Vorgehensmodell dazu beiträgt, dass tatsächlich das erwartete System entwickelt wird. Da sich die Anforderungen im Laufe des Projekts ändern können, entspricht ein basierend auf den zu Beginn des Projekts erhobenen Anforderungen entwickeltes System nicht zwangsweise den Erwartungen.

Anpassbarkeit Unter Anpassbarkeit verstehen die Autoren die Fähigkeit des Vorgehensmodells, sich an die individuelle Situation des Projekts anzupassen. Dieses Kriterium untersucht also die „Fähigkeit zum Tailoring“ (vergleiche Abschnitt 2.2.2).

Philosophie Die Autoren betrachten mit diesem Kriterium „das zugrunde liegende Paradigma, das primäre Ziel und das antreibende Moment des Vorgehensmodells“ [FK07].

Prozesssteuerung Mit diesem Kriterium beurteilen die Autoren von [FK07] das Vorgehensmodell zwischen den Polen Artefaktorientierung und Aktivitätsorientierung (vergleiche auch Abschnitt 2.3.2).

Benutzerbeteiligung Die Einbeziehung von Kunden und Endnutzern in die Erarbeitung des zu erstellenden Systems unterscheidet sich von Vorgehensmodell zu Vorgehensmodell stark. Der Grad an Beteiligung der Nutzer des zu entwickelnden Systems wird mit diesem Kriterium gemessen.

Die Autoren in [FK07] untersuchen mit diesen Kriterien *universelle Modelle*, namentlich das Wasserfallmodell, evolutionäre Ansätze, prototypische und nebenläufige Vorgehensmodelle, komponentenbasierte Entwicklung und formale Systementwicklung. Wie erwähnt, sind derartige Modelle in ihrer Beschreibung zu „roh“, als dass sich Aussagen über ihre Fähigkeit zur Anpassung machen ließen. *Universelle Modelle* betrachten wir bei der Entwicklung des feature-basierten Tailorings nicht weiter. Als Vertreter *vollständig* definierter Vorgehensmodelle (in [FK07] „weltliche Modelle“ genannt) betrachten die Autoren das V-Modell XT, den Rational Unified Process und Extreme Programming.

Der in [FK07] aufgestellte Kriterienkatalog zur Kategorisierung kann eine gute Ausgangsbasis für die Wahl eines geeigneten Vorgehensmodells sein. Für die Einschätzung der einzelnen Vorgehensmodelle anhand der Kriterien sei auf [FK07] verwiesen.

Die Kriterien zur Kategorisierung und deren beispielhafte Anwendung auf ausgewählte Vorgehensmodelle in [FK07] sind ein wichtiger Beitrag für die Auswahl von Vorgehensmodellen. Allerdings beziehen sich die vorgeschlagenen Kriterien in erster Linie auf das Vorgehensmodell selbst. Für die fundierte Wahl – und insbesondere für die projektspezifische Anpassung – eines Vorgehensmodells müsste diesen Kriterien ein Kriterienkatalog zur Charakterisierung der anwendenden Organisation oder des Projekts gegenübergestellt werden.

Der folgende Beitrag macht einen Vorschlag für die Wahl eines Vorgehensmodells anhand der Gegenüberstellung von Kriterien für Projekte und Vorgehensmodelle.

Eignungseinstufung von Vorgehensmodellen

In [Zie08] wurde eine Systematik zur Eignungseinstufung von Vorgehensmodellen entwickelt. Dazu wird ein Softwareentwicklungsprojekt anhand eines Katalogs von Projektmerkmalen beurteilt und das Ergebnis mit einer Menge von Vorgehensmodellmerkmalen abgeglichen.

Ein Projekt wird in [Zie08] charakterisiert durch eine Menge von Merkmalen aus den „Projektthemenbereichen“ Domänenwissen, Softwareentwicklung, Projektmanagement und Menschenführung. Weiterhin wird das Projekt hinsichtlich folgender „Risikobereiche“ eingeschätzt: Entwicklungs-Team, Entwicklungs-Produkt und Entwicklungs-Umfeld. Eine analoge Einschätzung wird für Vorgehensmodelle vorgenommen. So wird ein Vorgehensmodell beispielsweise hinsichtlich der Frage bewertet, wie gut es die Erhebung von Anforderungen unterstützt.

Mit der Charakterisierung des Projekts auf der einen Seite und den Profilen der Vorgehensmodelle auf der anderen kann ein Abgleich zwischen diesen erfolgen. Auf Basis dieses Abgleichs ist eine Entscheidung für bzw. gegen ein bestimmtes Vorgehensmodell möglich. Die Frage, ob und wie gut ein Vorgehensmodell die vier betrachteten Themenbereiche adressiert, erscheint allerdings nicht ausreichend für die Beurteilung und Wahl eines Vorgehensmodells. Die Projektthemenbereiche sind geeignet für eine grundsätzliche Verständigung auf eine *Philosophie* im Projekt (agile Vorgehensweise oder planorientiertes Vorgehensmodell, Grad und Umfang der zu erstellenden Dokumentation, usw.). Auch erscheint die in [Zie08] vorgenommene Einschätzung des Beitrags konkreter Vorgehensmodelle zum jeweiligen Projektthemenbereich subjektiv.

Bezug zur Arbeit

Beide diskutierten Beiträge sind hilfreich für die Einordnung und Wahl von Vorgehensmodellen. Beide Beiträge haben gemein, dass die betrachteten Vorgehensmodelle als „gegeben“ und unveränderbar betrachtet werden.

Durch das flexible Tailoring eines Vorgehensmodells kann ein und dasselbe Vorgehensmodell für unterschiedliche Projektsituationen maßgeschneidert werden. Insofern können die in [Zie08] entwickelte Methode zur Wahl eines Vorgehensmodells und der in [FK07] vorgeschlagene Kriterienkatalog ein vorbereitender Schritt für das feature-basierte Tailoring sein.

2.3.2 Aktivitäts- und Artefaktororientierung

In [MF11] wird unterschieden zwischen aktivitäts- und artefaktororientierten Vorgehensweisen. Zwischen diesen beiden Polen spannt sich ein Spektrum, auf welchem Vorgehensweisen eingeordnet werden können. Während die Ansiedlung einer Vorgehensweise auf diesem Spektrum nicht die Entscheidung für oder wider eine Vorgehensweise begründen kann, ist die Betrachtung beider Pole zur Einschätzung der Eignung einer Vorgehensweise für das feature-basierte Tailoring wertvoll.

Aktivitätsorientierung

Aktivitätsorientierte Vorgehensmodelle legen den Schwerpunkt auf die Beschreibung, *wie* ein Entwicklungsprojekt durchzuführen sei. Etwas vereinfacht wird

2.3 Einordnung von Vorgehensmodellen

davon ausgegangen, dass bei korrekter Durchführung des *Wie* am Ende auch die Ergebnisse (das *Was*) in der erwarteten Qualität vorliegen. Mit anderen Worten: Die gewünschte Form und Qualität der Ergebnisse soll sich aus der korrekten Umsetzung der Beschreibung, *wie* diese zu erstellen sein, ergeben.

Mit der Aktivität im Zentrum der Betrachtung (vergleiche Abbildung 2.7) ergibt sich der zeitliche Ablauf im Projekt als die Abfolge von Einzelaktivitäten.

Laut [MF11] beinhalten viele aktivitätsorientierte Ansätze überhaupt kein oder ein nur oberflächlich beschriebenes Artefaktmodell. Selbst wenn Artefakte beschrieben seien, so nur als Ergebnis einer Aktivität.

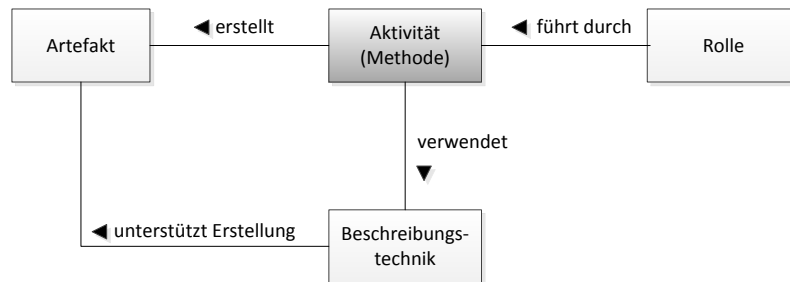


Abbildung 2.7: Aktivitätsorientierung (angelehnt an [MF11])

Artefaktorientierung

Im Zentrum artefakt- oder ergebnisorientierter Vorgehensmodelle liegt die Beschreibung der Ergebnisse (siehe Abbildung 2.8). Es wird im Wesentlichen beschrieben, *was* am Ende heraus kommen soll. Die Beschreibung von Aktivitäten, also *wie* ein Artefakt zu erstellen ist, beschränkt sich oft auf Verweise auf entsprechende Methoden oder Notationsformen.

Mit dem Artefakt im Zentrum der Betrachtung ergibt sich der zeitliche Ablauf im Projekt implizit aus den Abhängigkeiten zwischen Artefakten: Wenn das Ergebnis klar definiert ist, ergibt sich der Weg dorthin aus den für das Ergebnis erforderlichen Teil- oder Eingabeergebnissen. Aktivitäten dienen damit im Wesentlichen als Planungsgrößen.

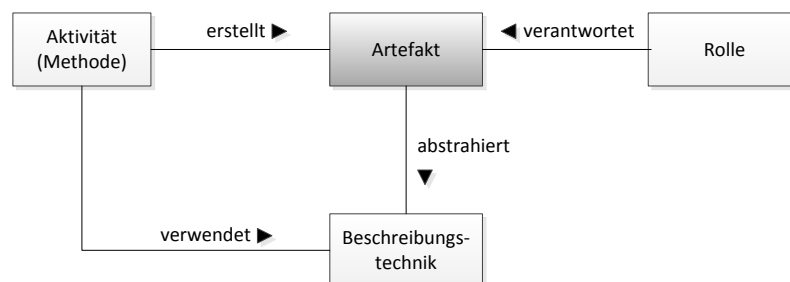


Abbildung 2.8: Artefaktorientierung (angelehnt an [MF11])

Bezug zur Arbeit

Wie in Abschnitt 2.2.1 ausgeführt, lassen sich in der aktuellen Literatur im Hinblick auf Vorgehensweisen bei Entwicklungsprojekten zwei Strömungen unterscheiden: die Begriffe *Vorgehensmodell* und *Methode* werden des Öfteren synonym verwendet - dabei fällt auf, dass (bei ansonsten gleicher Zielsetzung) Vorgehensmodelle in der Regel *eher* einer artefaktorientierten Philosophie folgen, während Methoden häufig *eher* aktivitätsorientiert ausgerichtet sind (vergleiche hierzu z.B. die Definition einer Methode nach [Bri96] in Abschnitt 2.2.1). In Abschnitt 2.2.1 haben wir für die vorliegende Arbeit eine Abgrenzung der beiden Begriffe vorgenommen.

Für die Anpassung bzw. Anpassungsfähigkeit und damit für das feature-basierte Tailoring macht es allerdings keinen Unterschied, welcher Philosophie ein Vorgehensmodell oder eine Methode folgt. In Kapitel 4 beleuchten wir eingehend, welche Voraussetzungen für die Umsetzung des feature-basierten Tailorings erfüllt sein müssen.

Da die Mechanismen des feature-basierten Tailorings erst dann ihre volle Wirkung entfalten, wenn das Ausgangsmodell genügend potenziell dem Tailoring unterworfenen bzw. zu unterwerfende Inhalte bietet, werden (eher) artefaktorientierte Vorgehensmodelle mit ihrer detaillierten Beschreibung der Arbeitsergebnisse mehr von den hier entwickelten Mechanismen profitieren.

2.3.3 Methodenneutralität

In [Ham08] wird unterschieden zwischen Vorgehensmodellen, die eine Methodik beinhalten und solchen, die unabhängig von einer Methode definiert sind. Die erste Klasse von Vorgehensmodellen wird *methodenspezifisch*, die zweite Klasse *methodenneutral* genannt.

Abbildung 2.9 veranschaulicht den Unterschied zwischen beiden Philosophien: der obere Teil der Abbildung stellt das in [Ham08] vorgeschlagene Integrationsmodell zwischen einem methodenneutralen Vorgehensmodell und einer Methodensammlung dar. Die Methodensammlung liegt *neben* dem Vorgehensmodell und wird von diesem referenziert. Die Verbindung zwischen Inhalten des Vorgehensmodells und denen der Methode erfolgt über die Zuordnung von Artefakten aus beiden Modellen.

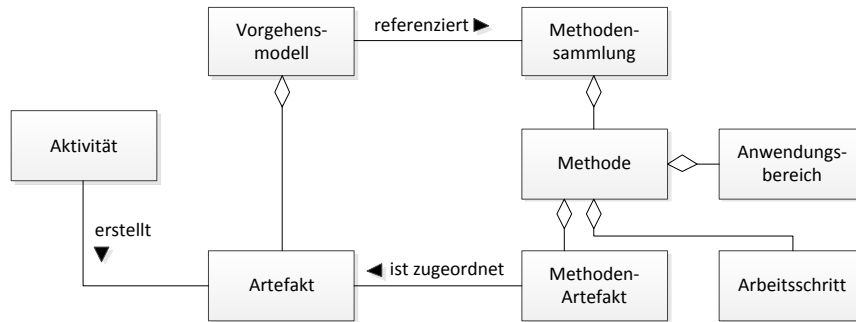
Der untere Teil der Abbildung stellt vereinfacht die Struktur eines methodenspezifischen Vorgehensmodells dar. Dieses beinhaltet implizit eine Methode – möglicherweise bestehend aus Teilmethoden für verschiedene Anwendungsbereiche. Die (Teil-)Methoden sind direkt in die Beschreibung der Artefakte und der Aktivitäten des Vorgehensmodells eingewoben.

Vertreter methodenspezifischer Vorgehen sind beispielsweise:

- die Durchführung von Softwareentwicklungsprojekten nach dem objektorientierten Paradigma,
- das Microsoft Solutions Framework [Tur06], da es die Verwendung einer bestimmten Werkzeugumgebung vorsieht, und
- in der Tendenz der Rational Unified Process [KK03] und der Open Unified Process [Ecl11b], da die Aktivitätsbeschreibungen dieser Vorgehensmodelle die Verwendung der UML vorsehen.

Methodenneutrale Vorgehensmodelle konzentrieren sich auf die Beschreibung der Projektorganisation und lassen die anzuwendenden Methoden offen. Sie

Methodenneutral



Methodenspezifisch

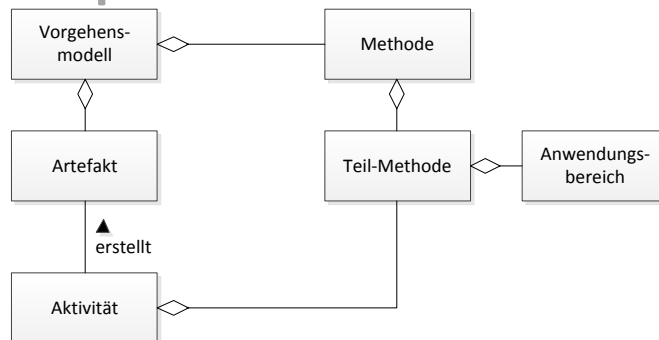


Abbildung 2.9: Methodenneutrale und methodenspezifische Vorgehensmodelle

sind damit potenziell in einer größeren Vielfalt von Projekten einsetzbar – bieten aber andererseits weniger konkrete Unterstützung für die täglichen Aufgaben im Projekt. Laut [Ham08] geschieht deshalb in der Regel folgendes:

Die Wahl geeigneter Methoden für ein Projekt erfolgt unabhängig vom Vorgehensmodell und liegt in den Händen des jeweiligen Projektleiters. Eine freie Methodenwahl bietet zwar maximale Flexibilität für die Methodenauswahl in den Projekten, es ist jedoch nicht gewährleistet, dass die im Rahmen der gewählten Methoden erarbeiteten Ergebnisse mit den vom Vorgehensmodell geforderten Ergebnissen zusammenpassen. Dies führt nicht selten zur (weitgehend als sinnlos erachteten) Nachdokumentation der geforderten Vorgehensmodellergebnisse parallel zur Erarbeitung der tatsächlichen Methodenergebnisse.

U. Hammerschall in [Ham08]

Durch den im oberen Teil von Abbildung 2.9 skizzierten Vorschlag der Integration von Methoden und Vorgehensmodellen wird dieser Bruch zwischen Vorgehensmodell und den im Projekt angewendeten Methoden überbrückt. Der Ansatz setzt voraus, dass in der Beschreibung der Methode klar definierte Artefakte vorliegen – dass also die Methodenbeschreibung *artefaktorientiert* ist. Wie auch in [MF11] ausführlich dargestellt, ist das für eine Vielzahl von Methodenbeschreibungen nicht der Fall.

Bezug zur Arbeit

Ein wesentlicher Schritt bei dem in [Ham08] vorgeschlagenen Ansatz der Methodenintegration ist die Auswahl geeigneter Methoden für die projektspezifische Ausprägung des Vorgehensmodells.

Da das vorgeschlagene Modell zur Methodenintegration unabhängig von einem konkreten Vorgehensmodell bleibt, sind die Aussagen zur projektspezifischen Anpassung und Methodenauswahl allgemein und abstrakt:

Die projektspezifische Anpassung findet entsprechend der vom Vorgehensmodell zur Verfügung gestellten Methodik statt. Ergebnis ist ein projektspezifisches Vorgehensmodell. Anschließend werden aus den im Rahmen der organisationsspezifischen Anpassung bereitgestellten Methodenmengen die für das jeweilige Projekt geeigneten Methoden ausgewählt.

U. Hammerschall in [Ham08]

In einer beispielhaften Anwendung des Integrationsmodells für Methoden auf das V-Modell XT werden *Projektmerkmale* des V-Modell XT (siehe auch Abschnitt 3.2.1) zur Auswahl von Methoden für die projektspezifische Ausprägung verwendet. In Abschnitt 3.2.1 und in der Fallstudie zum Tailoring des V-Modell XT in Abschnitt 7.3 diskutieren wir die Einschränkungen, die dieser Mechanismus zur Methodenauswahl mit sich bringt.

Wir greifen die Methodenintegration in den Ausführungen zur Umsetzung des feature-basierten Tailorings in Abschnitt 6.3.6 wieder auf. Werden das feature-basierte Tailoring und die Konzepte aus [Ham08] kombiniert, so ergibt sich ein mächtiges Werkzeug zur projektspezifischen *Spezialisierung in der Tiefe* (siehe auch Abschnitt 2.2.2).

2.4 Charakterisierung von Projekten

Im folgenden Abschnitt geben wir einen Überblick über Arbeiten zur Charakterisierung von Projekten und den daraus folgenden Strategien zur projektspezifischen Anpassung eines Vorgehensmodells. Im ersten Teil des Abschnitts befassen wir uns mit einem Ansatz, der sich mit der Ableitung von Anpassungszielen aus quantitativen Kennzahlen befasst. Im zweiten Teil des Abschnitts widmen wir uns Arbeiten, die Kriterien(-kataloge) zur Charakterisierung von Projekten vor dem Hintergrund der projektspezifischen Anpassung entwickelt haben.

2.4.1 Quantitatives Tailoring

In [BR87] wird festgestellt, dass in Softwareentwicklungsprojekten oft verlässliche und belastbare Kriterien fehlen für:

- die Wahl eines Vorgehensmodells und geeigneter Methoden.
- die Beurteilung der Qualität und Eignung der eingesetzten Vorgehensweise.
- die Verbesserung und Optimierung der eingesetzten Vorgehensweise.

Zur Lösung dieses Problems schlagen die Autoren die Verwendung der Anzahl und Art von Fehlern in Projektergebnissen vor. Diese seien messbar und würden unmittelbar mit Projektzielen korrelieren. Projektziele ließen sich durch geeignete Metriken über den Fehlern ausdrücken. Auch das Umfeld des Projekts

2.4 Charakterisierung von Projekten

ließe sich charakterisieren durch die Art und Anzahl der Fehler, die dieses im Projekt verursacht. Methoden wiederum ließen sich durch die Art und Anzahl der Fehler klassifizieren, die mit ihrer Verwendung einhergehen.

Wären die Anpassungsziele, die Vorgehens- und Methodenalternativen und äußere Einflüsse auf das Projekt derart quantifiziert, bestünde die Wahl und Anpassung geeigneter Methoden im Wesentlichen in der Optimierung der Fehlermetrik.

Allerdings: „The effectiveness of the improvement methodology depends on the amount of knowledge we have on the impact of methods and tools on defect profiles. Unfortunately, we do not have enough such knowledge yet. Most of the available knowledge is extremely environment dependent.“ [BR87]

Bezug zur Arbeit

[BR87] kann als Beitrag zur Entscheidungsunterstützung bei der Wahl geeigneter Methoden und bei deren Anpassung verstanden werden. Wie in Abschnitt 1.6 angedeutet, verstehen wir die Konzepte der vorliegenden Arbeit nicht als Beitrag zur Entscheidungsunterstützung beim Tailoring. Die Ideen aus [BR87] wären also eine Möglichkeit, Entscheidungen bei der Durchführung des feature-basierten Tailorings durch empirische Daten zu fundieren.

2.4.2 Tailoring-Kriterien

In [XR08] wird eine empirische Studie zur Ermittlung der Herausforderungen in Softwareentwicklungsprojekten beschrieben. Aus den Herausforderungen leiten die Autoren Anforderungen an das Tailoring eines Vorgehensmodells ab. Die in der empirischen Studie betrachteten Kriterien sind in Tabelle 2.1 zusammengefasst.

Für jede Kategorie der Kriterien in Tabelle 2.1 wurden in der Studie Strategien entwickelt, wie das Tailoring eines Vorgehensmodells zur Bewältigung der entsprechenden Herausforderungen beitragen kann. Tabelle 2.2 fasst einige der identifizierten Strategien zusammen – diese sollten jeweils Anwendung finden, wenn die entsprechende Kategorie als Herausforderung im Projekt aufgefasst wird, d.h. wenn die Kriterien in dieser Kategorie nicht „zufriedenstellend“ beantwortet werden können. Durch Anwendung der Tailoring-Strategien soll das Vorgehensmodell besser geeignet sein, den Herausforderungen im Projekt zu begegnen.

Da die Studie unabhängig von realen Vorgehensmodellen bleibt, finden sich darin keine Aussagen, wie die Strategien im konkreten Fall umzusetzen wären.

In [Coc00b, Coc06] wird für die Wahl einer Methode aus der Familie der Crystal Methodologies [Coc06, Coc04] die Charakterisierung des Projekts anhand der Kriterien *Kommunikationsaufwand*, *Kritikalität* und *Priorisierung* empfohlen.

Der Kommunikationsaufwand wird festgemacht an der Anzahl der Projektbeteiligten.

Mit Kritikalität des Projekts wird die Auswirkung von Fehlern im zu entwickelnden System auf die Umwelt eingeschätzt.

Durch die Priorisierung wird festgehalten, ob das Projekt juristische Konsequenzen haben kann (die Folge im Vorgehensmodell sollten Vorgaben für detaillierte und gründliche Dokumentation sein) oder ob Produktivität und Flexibilität im

Kategorie	Kriterium
Ressourcen	Angemessenheit des Projektbudgets
	Angemessenheit der Projektlaufzeit
	Verfügbarkeit von Personal mit den erforderlichen Fähigkeiten
	Verfügbarkeit der erforderlichen Werkzeuge
	Verfügbarkeit der erforderlichen technischen Unterstützung
Kommunikation	Bekanntheit der Teammitglieder aus vergangenen Projekten
	Zusammenarbeit im Team
	Größe des Teams
	Fluktuation im Team
	Projektdauer
	Hintergrund und Wissen der Akteure
Anforderungsmanagement	Klarheit und Eindeutigkeit des Projektvorschlags
	Anzahl der Interessensgruppen am Projekt
	Verfügbarkeit von Stakeholdern
	Erfahrung der Anwender des Systems mit der Domäne und dem System selbst
	Verfügbarkeit von Altsystemen oder Dokumenten
	Klarheit und Verständlichkeit der mit dem System verfolgten Geschäftsziele
	Vertrautheit der Anforderungsanalytiker mit der Geschäftsdomäne
	Stabilität der Anforderungen
Politik	Unterstützung der Geschäftsleitung
	Klarheit der Geschäftsziele des Projekts
	Verfügbarkeit der Geschäftsleitung
Technik	Vertrautheit des Teams mit den Werkzeugen
	Vertrautheit des Teams mit der verwendeten Technologie
	Vertrautheit des Teams mit dem Vorgehensmodell
	Verfügbarkeit technischer Unterstützung

Tabelle 2.1: Herausforderungen in Softwareentwicklungsprojekten

2.4 Charakterisierung von Projekten

Kategorie	Kriterium
Ressourcen	Reduktion der Aktivitäten, Dokumentation und Prozessschritte
	Reduktion des Detaillierungsgrads der Dokumentation
	Reduktion des „Zeremoniells“ (vgl. 3.1)
	Anforderungen des Prozesses durch Verhandlung mit den Stakeholdern reduzieren
Kommunikation	Vermehrung der formalen Dokumentation
	Häufigere formelle und informelle Meetings
	Entfernung von Dokumenten, die nicht zum Profil der Beteiligten passen
	Ersetzung von Dokumenten
Anforderungsmanagement	Mehr Iterationen mit Schwerpunkt Anforderungserhebung
	Häufigere Treffen mit Endnutzern
	Mehr formale Dokumente zum Anforderungsmanagement
	Mehr Feedback durch informelle Treffen und Fokusgruppen
Politik	Projektziele zu Beginn klar vereinbaren und kommunizieren
	Formale Dokumentation von Entscheidungen und Anforderungen
	Klare Definition der Geschäftsziele
Technik	Technologien evaluieren
	Trainings durchführen
	Technische Details dokumentieren
	Testaufwand auf technische Herausforderungen konzentrieren
	Anzahl der Iterationen erhöhen

Tabelle 2.2: Strategien zur Bewältigung der Herausforderungen

Projekt im Vordergrund stehen (mit der Folge, dass das Vorgehensmodell weniger Vorgaben hinsichtlich der Dokumentation macht).

Wie in Abbildung 2.10 schematisch zu erkennen, werden die drei Kriterien jeweils auf einer Achse aufgetragen. Je nach Einschätzung der Kriterien ergibt sich eine Zelle in der Matrix, welche die zu verwendende Variante der Crystal Methodenfamilie anzeigt.

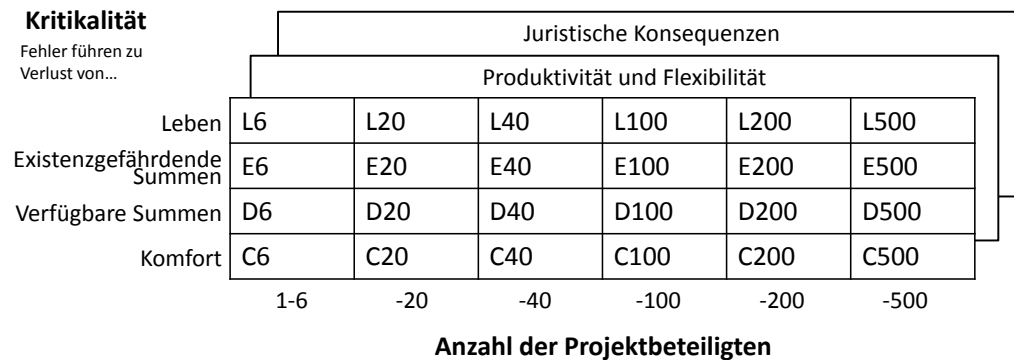


Abbildung 2.10: Kriterien zur Wahl einer Methode aus der Crystal Familie [Coc06]

Aus der Crystal Familie sind bisher nur zwei Varianten vollständig ausformuliert: Crystal Clear [Coc04] für kleine Teams (C6 - für 3 - 6 Mitarbeiter) und Crystal Orange [Coc06] für mittlere Teams (D40 - bis 40 Mitarbeiter).

In der Literatur zum *Situational Method Engineering* finden sich zahlreiche Arbeiten zur Charakterisierung von Projekten [Har97, tV97, HLW95]. Die dort genannten Kriterien zur Charakterisierung der Projektsituation sind in den Kriterienkatalog in Anhang A eingeflossen, weswegen wir sie hier nicht rekapitulieren. Gemein ist den Kriterienkatalogen, dass die Charakteristika und deren Wirkung auf projektspezifische Ausprägungen eines Vorgehensmodells informell beschrieben sind und mögliche Wechselwirkungen zwischen Kriterien unklar bleiben.

Besondere Erwähnung verdient der in [HLW95] beschriebene *Situation-Success-Scenario* (S³) Ansatz: Kern der Überlegung ist hier, dass erst die gemeinsame Betrachtung von Situationsfaktoren und Erfolgskriterien für das Projekt eine Methodenauswahl zulässt. Abbildung 2.11 stellt den Zusammenhang zwischen Projektsituation, Projekterfolg und den daraus resultierenden Aspekten schematisch dar. Das Szenario und die das Szenario bestimmenden Aspekte beschreiben in der Terminologie von [HLW95] die Vorgaben und Methoden, die im Projekt umzusetzen sind.

Bezug zur Arbeit

In allen bekannten Vorgehensmodellen und Methodensammlungen wird die Notwendigkeit zur projektspezifischen Anpassung anerkannt. Wie in Abschnitt 3.2.2 ausgeführt, finden sich aber beispielsweise bei EPF (und SPEM) keine Aussagen, nach welchen Kriterien und mit welchen Strategien das Tailoring durchgeführt werden sollte.

Da die Studie in [XR08] unabhängig vom Vorgehensmodell bleibt, gibt es hier keine Aussagen, wie und mit welchen Mitteln die vorgeschlagenen Strategien zur Bewältigung von Herausforderungen im Projekt umzusetzen wären.

2.4 Charakterisierung von Projekten

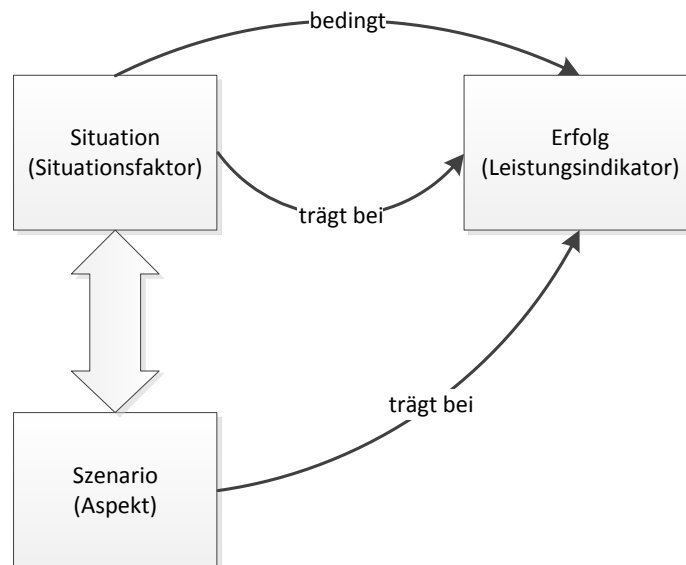


Abbildung 2.11: Der S³ Ansatz [HLW95]

Aus der in [Coc00b, Coc06] vorgeschlagenen Matrix (in Abbildung 2.10) ist zwar deutlich erkennbar, wie eine Charakterisierung und daraus folgend die Wahl einer Vorgehensweise erfolgen sollte. Da bisher aber erst zwei Mitglieder aus der Crystal Methodenfamilie ausformuliert sind, ist es schwer einzuschätzen, welche Auswirkungen die Einstufung eines Projekts in der Matrix für das Vorgehensmodell bzw. die Methode konkret haben soll.

Der *Situation-Success-Scenario* Ansatz aus [HLW95] schlägt neben einer Berücksichtigung der Situation des Projekts eine Einbeziehung der Erfolgskriterien vor. Wir haben derartige Kriterien zwar nicht in den Kriterienkatalog in Anhang A aufgenommen und in keiner der Fallstudien in Kapitel 7 berücksichtigt – prinzipiell spricht aber nichts dagegen, Erfolgskriterien beim feature-basierten Tailoring zu berücksichtigen.

Unter anderem inspiriert von der Studie in [XR08] beschreiben wir im Folgenden eine im Rahmen einer Forschungs Kooperation zwischen der T-Systems International GmbH, den Telekom Laboratories und der Technischen Universität durchgeführte Studie zur Ermittlung des Bedarfs für projektspezifische Anpassung des von der T-Systems International GmbH hausintern verwendeten Vorgehensmodells. Der in der Studie verwendete Projektsteckbrief (in Tabelle 2.3) bietet ebenfalls Kriterien zur Charakterisierung von Projekten und ist in den Katalog im Anhang A eingeflossen.

2.4.3 Studie zur Anwendung eines Vorgehensmodells

Um die Anwendung, Akzeptanz und den Bedarf für Anpassung von Vorgehensmodellen zu ermitteln, haben wir eine Befragung von Vorgehensmodell-Anwendern durchgeführt. In diesem Abschnitt geben wir einen kurzen Überblick über den Studienaufbau, die Durchführung und die Ergebnisse.

Ziele der Studie

Ziel der Studie war die Ermittlung des Bedarfs für Anpassung des SE Book. Dieses ist ein ausgereiftes Vorgehensmodell, welches bei der T-Systems International GmbH hausintern eingesetzt wird. Zur Ermittlung dieses Bedarfs sollte die Studie folgende Fragen beantworten:

- *Wie* wird das Vorgehensmodell derzeit in der Praxis eingesetzt?
- *Welche Teile* des Vorgehensmodells werden in der Praxis (überwiegend) verwendet?
- *Welche Aspekte des Projektalltags* sind vom Vorgehensmodell bisher nicht oder nicht zur Genüge berücksichtigt?
- Wird die Anwendung des Vorgehensmodells als *hilfreich oder hinderlich* im Projektalltag empfunden?
- Ist eine *Korrelation* erkennbar zwischen Projekten mit bestimmten Eigenschaften und der Anwendung des Vorgehensmodells?

Studienaufbau und -durchführung

Zur Beantwortung der in der Zielstellung der Studie formulierten Fragen wurden alle zur Verfügung stehenden Quellen ausgewertet (siehe Abbildung 2.12):

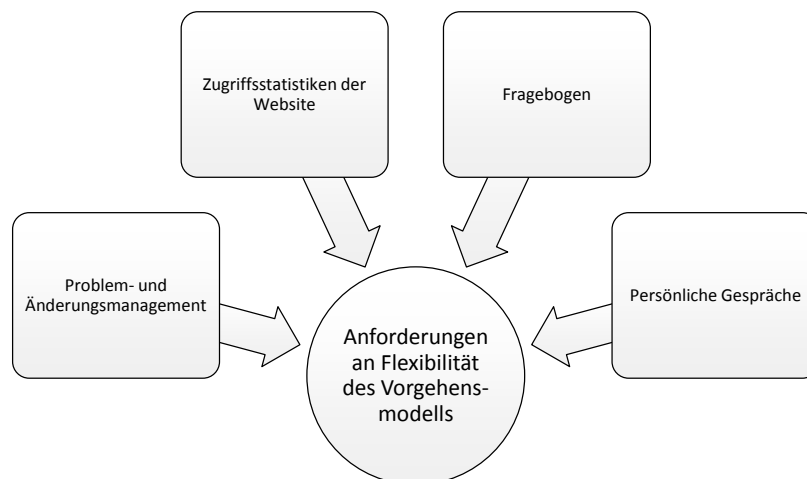


Abbildung 2.12: Untersuchungsgegenstände der Studie

Problem- und Änderungsanträge Zu dem untersuchten Vorgehensmodell gibt es ein Problem- und Änderungsmanagement, dessen Datenbank zum Zeitpunkt der Studie insgesamt 288 Einträge enthielt. Ein Großteil der Einträge entfiel auf redaktionelle Details, wie z. B. zu ändernde Formulierungen oder zu überarbeitende grafische Darstellungen. In der Studie wurden nur solche Problemmeldungen oder Änderungsanträge berücksichtigt, die

- einen *direkten Zusammenhang* mit dem Tailoring des Vorgehensmodells erkennen lassen.
- Hinweise und Anhaltspunkte geben für *Bedarf an einer Flexibilisierung* des Vorgehensmodells.
- Anforderungen an das Vorgehensmodell stellen, die nicht allgemeingültig sind sondern *Teil einer speziellen Ausprägung* sein müssten.

2.4 Charakterisierung von Projekten

Zugriffsstatistiken der Webseite Zu dem untersuchten Vorgehensmodell gibt es eine offizielle Webseite, welche die Dokumentation der verschiedenen Versionen des Vorgehensmodells beinhaltet. Diese Webseite ist eine der zentralen Anlaufstellen für Informationen zum SE Book. Eine Auswertung der am häufigsten zugegriffenen Inhalte und der am häufigsten verwendeten Suchbegriffe gab Aufschluss über die für die Anwender relevanten Inhalte. Um aussagekräftige Ergebnisse zu erhalten, wurden die IP-Adressen der Abteilung, die das Vorgehensmodell entwickelt und betreut, aus der Betrachtung ausgeschlossen. Der betrachtete Zeitraum betrug 3 Monate zwischen Februar 2010 und Mai 2010.

Fragebogen Zusammen mit den Partnern bei T-Systems und bei den Telekom Laboratories wurde ein Fragebogen entwickelt, der von Anwendern des Vorgehensmodells (Projektleitern oder unternehmensinternen Beratern) beantwortet wurde.

Persönliche Gespräche Basierend auf den Rückläufern des Fragebogens wurden mit den Teilnehmern der Umfrage persönliche Gespräche durchgeführt.

Der Fragebogen hatte das Ziel, einen Zusammenhang zwischen Charakteristika der Projekte (beispielsweise Anzahl der Mitarbeiter im Projekt oder Projektgegenstand) und der Anwendung des Vorgehensmodells zu ermitteln. Im Einzelnen war der Fragebogen wie folgt aufgebaut:

Im ersten Teil wurden Charakteristika des Projekts abgefragt. War ein Teilnehmer der Umfrage zum Zeitpunkt der Befragung in mehreren Projekten involviert, so wurde er gebeten, den Fragebogen entweder für jedes Projekt individuell auszufüllen, oder aber *ein konkretes* Projekt zu beschreiben. Bei den abgefragten Charakteristika haben wir uns an einem unternehmensintern bereits existierenden Projektsteckbrief orientiert. Tabelle 2.3 gibt einen Eindruck über die in diesem Teil des Fragebogens gestellten Fragen.

Im zweiten Teil wurden allgemeine Aussagen zum Vorgehensmodell abgefragt. Der dritte Teil orientierte sich strukturell an den Submodellen eines Vorgehensmodells (vergleiche Abschnitt 2.2.1): Abgefragt wurden Aussagen zum Artefaktmodell, zum Aktivitätsmodell, zum Rollenmodell und zu den vom Vorgehensmodell zur Verfügung gestellten Vorlagen für Ergebnisse.

Um eine statistische Auswertung des Fragebogens zu ermöglichen, haben wir die Fragen des zweiten Teils so formuliert, dass sie durch Antworten auf einer Likert-Skala [Lik32] beantwortet werden konnten. Neben diesen geschlossenen Fragen konnte jeder Teilnehmer der Umfrage in einem Freitextfeld die Antwort erläutern oder Stellung beziehen. Tabelle 2.4 zeigt exemplarisch den Aufbau des Fragebogens für den Teilbereich des Artefaktmodells.

In der zweiten Spalte ist jeweils die Art der erwarteten Antwort angegeben:

- Bei einer *Freitext*-Frage konnte der Teilnehmer beliebige Angaben machen, beispielsweise spezielle Artefakte benennen.
- Bei einer *Zustimmung*-Frage konnte der Befragte seine Zustimmung zur formulierten Frage auf der folgenden Skala angeben: *stimmt gar nicht*, *stimmt wenig*, *stimmt teils-teils*, *stimmt ziemlich*, *stimmt völlig*.
- Bei einer *Auswahl*-Frage wurde dem Teilnehmer eine Liste der vom Vorgehensmodell geforderten Artefakte ¹ präsentiert.

¹ und bei den Abschnitten zu den anderen Submodellen entsprechende Listen.

Frage	beispielhafte Belegung
Projektkategorie	Unternehmensinterne RisikoEinstufung des Projekts, im Wesentlichen abhängig vom Projektvolumen
Anzahl Mitarbeiter im Projekt	20
Projektlaufzeit	24 Monate
Anforderungen bezüglich Safety/Security	niedrig, mittel, hoch
Art des Projekts	Neuentwicklung, Beratung, Anpassung, etc.
Fachliche Lösung	CRM System, ERP System, Sonstige, etc.
Technische Lösung	Web-Anwendung, Batchsystem, Dialogsystem, etc.
Relevante Phasen	Spezifikation, Systementwurf, Programmierung, Integration, etc.
Vorgehensmodell des Kunden	V-Modell XT, gefordertes CMMI Level, etc.

Tabelle 2.3: Projektsteckbrief des Fragebogens

In der dritten Spalte der Tabelle 2.4 ist angegeben, welcher Bedarf für Anpassung des Vorgehensmodells aus der Frage abgeleitet werden kann. Hierbei haben wir uns an [XR08] orientiert: Für das Tailoring werden dort folgende grundlegende Operationen auf Inhalten eines Vorgehensmodells identifiziert:

Add Das Element ist im Ziel-Vorgehensmodell noch nicht vorhanden und sollte hinzugefügt werden.

Downsize Das Element ist zu umfangreich für die Anforderungen des Projekts. Der Umfang (beispielsweise der Beschreibung) sollte reduziert werden.

Drop/Skip Das Element wird im Projekt nicht benötigt und sollte aus der projektspezifischen Ausprägung entfernt werden.

Expand/Concretize Das Element ist für die Erfordernisse des Projekts nur unzureichend beschrieben. Es sollte detailliert werden.

Redefine Das Element ist im Vorgehensmodell vorhanden, hat aber im Projekt eine andere Bedeutung. Es muss für die Erfordernisse des Projekts umdefiniert werden.

Replace Das vorhandene Element im Vorgehensmodell muss in der projektspezifischen Ausprägung durch ein anderes Element ersetzt werden.

Ziel des Fragebogens war, eine *Korrelation* zwischen den im Projektsteckbrief abgefragten Projektcharakteristika und den Aussagen zu Inhalten der einzelnen Submodelle des Vorgehensmodells zu ermitteln.

2.4 Charakterisierung von Projekten

Frage	Skala	Aussage
Die Ergebnisstruktur ist teilweise zu umfangreich für unser Projekt.	Zustimmung	Drop/Skip
Die Ergebnisse sind teilweise nicht ausreichend für unser Projekt.	Zustimmung	Add
Viele Ergebnisse werden nicht einzeln erstellt, sondern fallen zusammen.	Zustimmung	Merge
Die Beschreibungen der Ergebnisse sind zu allgemein und müssen für die speziellen Gegebenheiten verfeinert werden	Zustimmung	Expand/Concretize
Nicht erstellte Ergebnisse (Priorität 1)	Auswahl	Drop/Skip
Nicht erstellte Ergebnisse (Priorität 2)	Auswahl	Drop/Skip
Nicht erstellte Ergebnisse (Priorität 3)	Auswahl	Drop/Skip
Zu allgemein beschriebene Ergebnisse (Priorität 1)	Auswahl	Expand/Concretize
Zu allgemein beschriebene Ergebnisse (Priorität 2)	Auswahl	Expand/Concretize
Zu allgemein beschriebene Ergebnisse (Priorität 3)	Auswahl	Expand/Concretize
Ergebnisse ohne Entsprechung im Vorgehensmodell	Freitext	Add
Im Projekt zusammenfallende Ergebnisse	Freitext	Merge

Tabelle 2.4: Fragebogen – Auszug Artefaktmodell

Ergebnisse der Studie

Zentrale Ergebnisse unserer Untersuchung waren:

Wichtigster Bestandteil des Vorgehensmodells sind seine Vorlagen Das SE Book bietet zu jedem Arbeitsergebnis detaillierte Vorlagen zur Erstellung. Zu den Vorlagen gibt es weiterhin sog. *Hilfsmittel*, die je nach Ergebnis den Charakter einer Checkliste oder einer Ausfüllhilfe haben. 25% der Zugriffe auf die Website des Vorgehensmodells entfallen auf Vorlagen und Ausfüllhilfen. 50% der Zugriffe fallen (erwartungsgemäß) auf die Einstiegsseite der Webseite. Auf die restlichen Inhalte des Vorgehensmodells – also die detaillierte Dokumentation des Vorgehensmodells – entfielen die verbleibenden 25% der Zugriffe.

Das Vorgehensmodell wird pragmatisch gelebt Eine der allgemeinen Fragen im Fragebogen war, ob aufgrund der Vorgaben des Vorgehensmodells zusätzliche Arbeiten im Projekt gemacht wurden. Diese Frage wurde überwiegend negativ beantwortet. Gleichzeitig wurde das Vorgehensmodell von den Teilnehmern der Umfrage als sehr detailliert und umfangreich eingeschätzt. Diese beiden Aussagen zusammen lassen den Schluss zu, dass vom Vorgehensmodell vorgegebene Aktivitäten und Ergebnisse nicht erarbeitet werden, wenn sie als nicht förderlich für die Erreichung der Projektziele erachtet werden.

Anforderungen an die Breite Viele Wünsche aus dem Problem- und Änderungsmanagement, aus der Umfrage und aus den persönlichen Gesprächen betrafen die *Breite* des Vorgehensmodells. Folgende Beispiele stehen stellvertretend für diese Klasse:

- Eine spezielle Ausprägung des Vorgehensmodells für Kleinstprojekte wird gewünscht.
- Eine spezielle Ausprägung des Vorgehensmodells für Beratungsprojekte wird gewünscht.
- Das SE Book sollte Vorgaben für die frühen Phasen von Projekten beinhalten, die nicht auf der grünen Wiese starten (z. B. Analyse eines Altsystems anstatt Anforderungserhebung für ein Neusystem).
- Die Vorgaben des SE Book sollten auf einige Phasen (im Sinne eines Wasserfalls) begrenzt werden können. Für ein Projekt, dessen Aufgabe nur die Erarbeitung der Anforderungen zusammen mit dem Kunden ist, sind große Teile eines kompletten Vorgehensmodells nicht relevant.
- Die Detaillierung und Verbindlichkeit der Vorgaben sollte abhängig von der Sicherheitskritikalität des zu entwickelnden Systems sein.

Anforderungen an die Tiefe Ebenso wurden zahlreiche Wünsche und Verbesserungsvorschläge geäußert, die mit einer *Vertiefung*, Konkretisierung und Spezialisierung einzelner Inhalte zu tun hatten. Beispiele hierfür waren:

- Spezielle Vorgaben zum Umgang mit Unterauftragnehmern oder Offshore-Projekten.
- Richtlinien für spezielle technische Lösungen (z. B. Richtlinien für SQL Datenbankprojekte).
- Abbildung der Inhalte des Vorgehensmodells auf US GAAP.
- Präzisere Vorgehensmodellinhalte, wenn das Projekt die Entwicklung einer Benutzerschnittstelle umfasst (Design, Benutzbarkeit, Fehlertoleranz der Dateneingabe, etc.).

2.4 Charakterisierung von Projekten

- Spezielle Ergebnisvorlagen für die existierende Werkzeuglandschaft im Unternehmen.

Diese Wünsche und Anforderungen legen eine Integration geeigneter Methoden in das Vorgehensmodell nahe.

Die Anwender pflegen einen pragmatischen Umgang mit dem Vorgehensmodell. Wo es in der Breite nicht passt, werden ganze Teile der Vorgaben ignoriert (das Problem ist hier nicht die Tatsache, dass Teile ignoriert werden, sondern dass Teile ignoriert werden müssen). Wo das Vorgehensmodell unzureichend Hilfestellung bietet (etwa für SQL Datenbankprojekte oder bei der Abbildung der Inhalte auf US GAAP), wird auf eigene, individuelle Lösungen ausgewichen.

Bewertung und vergleichbare Untersuchungen

Die Studie wurde in *einem* großen deutschen Unternehmen für *ein* Vorgehensmodell durchgeführt. Die Ergebnisse sind aber größtenteils unabhängig vom konkreten Vorgehensmodell und zeichnen ein ähnliches Bild wie vergleichbare Untersuchungen.

In [FK07] wird eine Umfrage zum Einsatz von Softwareentwicklungs-Vorgehensmodellen, Reifegradmodellen, verteilter Entwicklung und Produktlinien vorgestellt (durchgeführt im Rahmen des Projekts IOSE-W² ²). Für die Details der Umfrage sei auf den Beitrag verwiesen. Zwei Ergebnisse wollen wir hier dennoch herausgreifen und diskutieren:

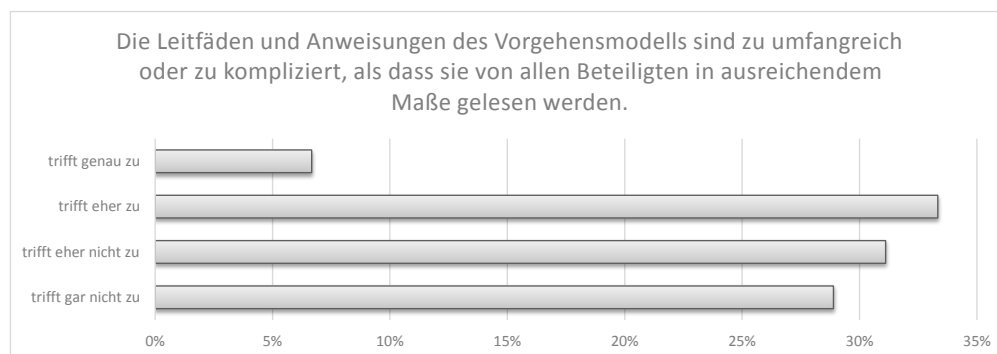


Abbildung 2.13: Ergebnis der Umfrage zum Einsatz von Vorgehensmodellen (1) (angelehnt an [FK07])

Laut Abbildung 2.13 haben 40% der Teilnehmer der Umfrage angegeben, dass die Leitfäden und Anweisungen des eingesetzten Vorgehensmodells zu umfangreich seien, als dass diese detailliert gelesen würden. Dieses Ergebnis deckt sich mit den Ergebnissen unserer eigenen Untersuchung und ist ein deutlicher Hinweis darauf, dass die Passgenauigkeit von Vorgehensmodellen gesteigert werden kann und sollte.

Auf die Frage nach der Akzeptanz des Vorgehensmodells gaben immerhin 22% der Befragten an, dass dieses von den Projektbeteiligten nicht als sinnvoll und hilfreich erachtet würde (vergleiche Abbildung 2.14). Zwar sagen nur halb so viele Befragte, dass das Vorgehensmodell nicht akzeptiert werde wie dass es nicht gelesen würde – es kann aber vermutet werden, dass die „Ignoranz“ des

² www.iose-w.de

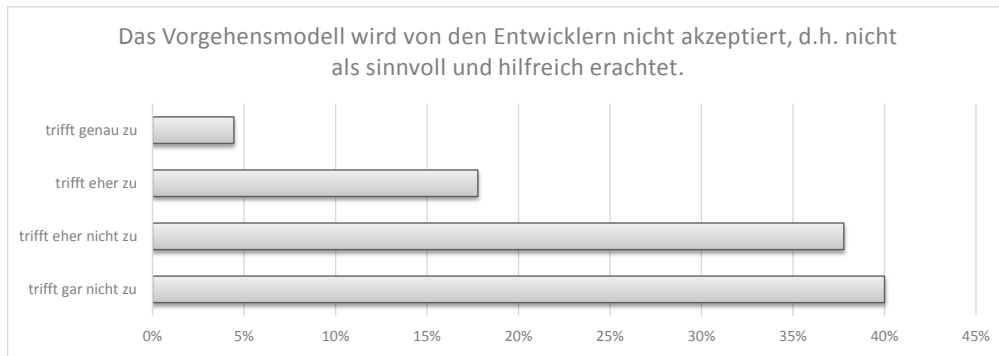


Abbildung 2.14: Ergebnis der Umfrage zum Einsatz von Vorgehensmodellen (2) (angelehnt an [FK07])

Vorgehensmodells auch als „Akzeptanz“ verstanden werden kann („so lange ich nichts damit zu tun habe, stört es mich nicht“). Beide Antworten zusammen genommen zeichnen wieder das auch in unserer Untersuchung festgestellte Bild eines eher pragmatischen Verhältnisses zum Vorgehensmodell.

Insbesondere bemerkenswert und keinesfalls niedrig erscheinen die 22%, wenn man die in der Umfrage betrachteten Vorgehensmodelle berücksichtigt. Wie aus Abbildung 2.15 hervorgeht³, entfällt ein großer Anteil der genannten Vorgehensmodelle auf sehr leichtgewichtige Vorgehensweisen oder *Methoden*, wie z. B. Object Oriented Design (OOD), Agile Modelling, Feature Driven Development (FDD) und Model Driven Architecture (MDA)⁴.

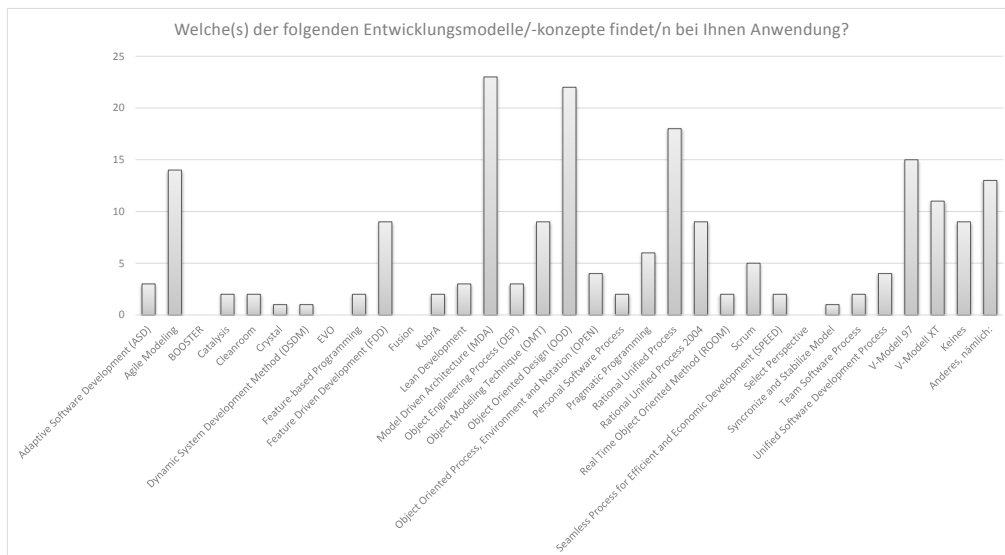


Abbildung 2.15: Ergebnis der Umfrage zum Einsatz von Vorgehensmodellen (3) (angelehnt an [FK07])

³ Die Höhe der Balken entspricht der Anzahl der Nennungen der jeweiligen Vorgehensweise durch die Teilnehmer der Umfrage.

⁴ in Abschnitt 2.2.1 haben wir die Begriffe *Vorgehensmodell* und *Methode* für den weiteren Verlauf der Arbeit voneinander abgegrenzt. Viele der in der Umfrage genannten „Vorgehensmodelle“ würden den Bedingungen, die wir für unsere Arbeit an ein Vorgehensmodell stellen, nicht genügen.

2.4 Charakterisierung von Projekten

In [KLS11] wurde eine Umfrage unter deutschen Behörden durchgeführt. Ziel der Untersuchung war es, einen Überblick zu gewinnen über die Verbreitung und den Einsatz des V-Modell XT in deutschen Behörden.

Auch hier greifen wir zwei Ergebnisse heraus:

Von sechs für eine vertiefende Befragung ausgewählten Behörden gab nur eine an, dass sie ein auf die Projektbedürfnisse zugeschnittenes V-Modell XT verwendet. Dieses Ergebnis ist überraschend, da die Anpassungsfähigkeit des V-Modell XT als eine seiner Stärken gilt (vergleiche hierzu auch 3.2.1).

Kriterium/Frage	Ergebnis
Verglichen mit Projekten ohne V-Modell XT trägt dieses bei zu:	
Kommunikation	leichte Verbesserung (0,35)
Mehraufwand	leichte Verschlechterung (-0,52)
Flexibilität	weder noch (0,04)

Tabelle 2.5: Ergebnis der Umfrage zum Einsatz des V-Modell XT (angelehnt an [KLS11])

Das zweite bemerkenswerte Ergebnis ist in Tabelle 2.5 dargestellt⁵: Auch in [KLS11] hat sich aus der Befragung der Anwender ergeben, dass die Anwendung des Vorgehensmodells Mehraufwand verursachen kann. Die Gründe hierfür können natürlich vielfältig sein und mögliche Interpretationen dieses Ergebnisses werden in [KLS11] angeboten. Beide Ergebnisse zusammen legen den Schluss nahe, dass ein besser auf die Organisation oder das Projekt angepasstes Vorgehensmodell den Mehraufwand reduzieren hätte können.

Bezug zur Arbeit

Die vorgestellte Studie zum SE Book ist selbst Teil und Beitrag der vorliegenden Arbeit. Sie stellt eine wesentliche Motivation für die Entwicklung des feature-basierten Tailorings dar. Auch wenn die Ausgestaltung projektspezifisch anpassbarer Inhalte abhängig vom konkreten Vorgehensmodell bleibt, schafft das feature-basierte Tailoring die nötigen Voraussetzungen, die in der Studie zutage getretenen Bedürfnisse hinsichtlich der Anpassbarkeit von Vorgehensmodellen zu befriedigen.

2.4.4 Von Projekteigenschaften zum Vorgehensmodell

Den Bedarf für Anpassung von Vorgehensmodellen haben wir in einer zweiten Studie [KK13] in Form einer systematischen Literaturrecherche aus einem anderen Blickwinkel untersucht. Das Ergebnis der Recherche ist ein Überblick über existierende Literatur zu Projekteigenschaften beim Tailoring von Vorgehensmodellen. Weiterhin fasst sie die Aussagen in der Literatur zusammen, *welche Wirkung* die Projekteigenschaften auf ein Vorgehensmodell haben (sollten).

⁵ Die Zahlen in Klammern in der rechten Spalte der Tabelle geben das durchschnittliche Ergebnis auf einer Skala von -2 (Verschlechterung) bis +2 (Verbesserung) an.

Ziele

Ziel der Studie war, einen Überblick zu erlangen über Aussagen in der Literatur zu folgenden Fragen:

Forschungsfrage 1: Welche Projekteigenschaften spielen beim Tailoring eines Vorgehensmodells eine Rolle bzw. können eine Rolle spielen?

Forschungsfrage 2: Welche Auswirkung auf die projektspezifische Ausprägung des Vorgehensmodells haben die Projekteigenschaften bzw. sollten diese haben?

Studien mit vergleichbarer Zielsetzung – wie zum Beispiel [XR08] (vergleiche Abschnitt 2.4.2) – beantworten diese Fragen in der Regel durch eine empirische Untersuchung. Mit der hier vorgestellten Studie sammeln wir diese Ergebnisse und kondensieren daraus einen Katalog von Kriterien zum Tailoring von Vorgehensmodellen.

Aufbau und Durchführung

Bei der Durchführung der Recherche haben wir uns an dem Vorgehen für *Systematic Literature Reviews* [KBB⁺09] orientiert. Konkret folgt der Aufbau der Untersuchung dem Vorschlag von Runeson in [RH09].

Die Forschungsfragen haben wir bereits im vorherigen Abschnitt 2.4.4 formuliert. Um Forschungsfrage 1 zu beantworten, haben wir ein *Systematic Literature Review* durchgeführt. Das Verfahren zur Identifikation relevanter Literatur wird im Folgenden genauer erläutert. Um Forschungsfrage 2 zu beantworten, haben wir die im ersten Schritt identifizierte Literatur eingehend analysiert.

Auswahl relevanter Literatur: Die Auswahl relevanter Literatur erfolgte durch eine automatisierte Suche in den folgenden Literaturdatenbanken:

- ACM Digital Library
- SpringerLink
- IEEE Computer Society Digital Library
- Wiley

Die Entscheidung für diese Datenbanken fiel nach internen Diskussionen und der Erfahrung der an der Recherche beteiligten Personen im Feld des Process Engineering. Die Erfahrung der Autoren führte auch zur Berücksichtigung von weiteren relevanten Werken in diesem Bereich (wie etwa [Inf04, KTF09, RHB⁺06] und dem SE Book der T-Systems International GmbH), auch wenn diese nicht in einer der untersuchten Datenbanken verzeichnet waren.

Zur Ermittlung geeigneter Suchbegriffe in den Datenbanken wurde eine „Saat“ initialer Beiträge verwendet. Diese Beiträge sind in Tabelle 2.6 zusammengefasst. Aus diesen Beiträgen wurden iterativ geeignete Suchbegriffe extrahiert und diese validiert, indem die jeweils resultierenden Suchergebnisse auf das Vorhandensein der initialen Referenzwerke hin untersucht wurden. Das Ergebnis dieses Verfahrens waren die folgenden Suchbegriffe:

- (software process tailoring)
- (method engineering)
- (process **or** method **or** methodology) **and** (tailoring **or** adaption **or** customization **or** customisation)
- process tailoring **and** (practice **or** experience **or** study)

2.4 Charakterisierung von Projekten

- software process **and** (standard **or** CMMI **or** ISO **or** SPICE) **and** compliance
- method **and** (engineering **or** weaving) **or** situational method engineering

Jeder der Suchbegriffe wurde auf alle vier Datenbanken angewendet. Aus jeder Datenbank haben wir die ersten 160 Treffer in die Ergebnismenge übernommen.

Autoren	Beitrag
Pedreira et. al.	[PPLB07]
Brinkkemper, Sjaak	[Bri96]
Harmsen et. al.	[Har97, HLW95]
Xu Peng	[XR08]
Méndez Fernández et. al.	[MFWL ⁺ 12]

Tabelle 2.6: Referenz-Beiträge zur Ermittlung der Suchbegriffe

Zu jedem Beitrag in der Ergebnismenge wurden die in Tabelle 2.7 dargestellten Daten erfasst.

Kategorie	Erfasste Daten
Metadaten	ID, Zitierungs-Schlüssel, Treffer bei welchem Suchbegriff
Inhalt	Autoren, Titel, Kurzfassung, Jahr
Abstimmung	Abstimmungsergebnis der Autoren über die Relevanz des Beitrags für die Untersuchung, Kommentare zur Abstimmung

Tabelle 2.7: Aufbau der Literatursammlung

Nach einer Harmonisierung der Ergebnismenge (insbesondere der Eliminierung doppelt gefundener Einträge) haben die Autoren der Studie zu jedem Beitrag abgestimmt, ob dieser für die weitere Betrachtung zu berücksichtigen ist. Waren sich beide Autoren in ihrem Urteil einig, wurde der Beitrag entsprechend übernommen oder ausgeschlossen. Fiel das Urteil der Autoren unterschiedlich aus, wurde der Beitrag einem dritten Experten vorgelegt, welcher die endgültige Entscheidung fällte.

Beantwortung der Forschungsfragen: Um zu beantworten, welche Kriterien in der Literatur als entscheidend für die projektspezifische Anpassung von Vorgehensmodellen betrachtet werden, haben wir die in die engere Auswahl gekommenen Beiträge eingehend untersucht, und darin erwähnte Kriterien in eine Kriterienliste aufgenommen. Dabei blieb es dem Urteil der Autoren überlassen, wann und ob ein Kriterium als Synonym für ein bereits aufgenommenes betrachtet werden konnte.

Um zu beantworten, welche Implikationen ein Kriterium für die projektspezifische Ausprägung eines Vorgehensmodells hat (bzw. haben sollte), wurde die Literatur für jedes gefundene Kriterium auf entsprechende Hinweise untersucht. Basierend auf diesen Hinweisen oder der Begründung für ein Kriterium („rationale“) haben wir *mögliche* Auswirkungen des Kriteriums auf ein Vorgehensmodell identifiziert.

Ergebnisse und Diskussion

Die Literatursuche in den einschlägigen Datenbanken, basierend auf den oben genannten Suchbegriffen, ergab nach Homogenisierung und Abstimmung über die Relevanz 127 Beiträge. Tabelle 2.8 zeigt die Zahl der gefundenen Artikel pro Datenbank (Spalte Anzahl), nach Bereinigung der mehrfach gefundenen Beiträge (Spalte Anzahl bereinigt) und nach Abstimmung über die Relevanz (Spalte Ausgewählt). Die Datenbank „Andere“ bezieht sich auf jene Quellen, die von den Autoren als relevant betrachtet wurden, auch wenn sie nicht in einer der Datenbanken verzeichnet waren.

Datenbank	Anzahl	Anzahl bereinigt	Ausgewählt
ACM	210	210	44
Springer	60	60	13
IEEE	210	210	22
Wiley	1120	329	44
Andere			4
Sum	1600	809	127

Tabelle 2.8: Statistik zur betrachteten Literatur

Die Ergebnisse in Bezug auf Forschungsfrage 1 befinden sich im Kriterienkatalog zum Tailoring in Anhang A. Viele (28) Kriterien befassen sich mit dem Umfeld des Projekts, etwa der Verfügbarkeit von Interessensgruppen, der Endnutzer und des Managements oder anderen Randbedingungen an das Projekt, wie etwa Budget und Vertragsart. Neun Kriterien beziehen sich auf das Projektteam, etwa dessen Größe und geographische Verteilung. Als letzte große Gruppe konnten 12 Kriterien identifiziert werden, welche sich im weitesten Sinne mit den Zielen des Projekts befassen. In diese Kategorie fallen beispielsweise der Innovationsgrad und die Komplexität des Projekts. Das in der Literatur am häufigsten genannte Kriterium zur Anpassung einer Vorgehensweise ist die Teamgröße (bzw. etwas weiter gefasst die Projektgröße) ([XR08, Coc06, CO08, HLW95, MFWL⁺12], SE Book).

Eine Reihe der Kriterien kann nicht klar voneinander abgegrenzt werden, oder die Kriterien weisen eine gewisse Überlappung auf. Dies sind insbesondere solche Projekteigenschaften, die einen Aspekt des Projekts aus unterschiedlichen Blickwinkeln betrachten. Verfügbarkeit und Unterstützung des Managements sind Beispiele für derartige Kriterien. Allerdings können diese nicht zusammengefasst werden, da sie die Umstände des Projekts in unterschiedlicher Weise charakterisieren und für die Vorgehensweise möglicherweise unterschiedliche Auswirkung haben. So ist eine gute Verfügbarkeit des Managements für das Projekt nur bedingt hilfreich, wenn dieses nicht hinter dem Projekt steht. Gleichzeitig resultiert aus Unterstützung nicht automatisch eine gute Verfügbarkeit. Andere Kriterien – wie etwa die Verfügbarkeit des Kunden und die Verfügbarkeit der Endnutzer – können in bestimmten Konstellationen zusammenfallen. Das Kriterium Verfügbarkeit von Stakeholdern wiederum kann aufgefasst werden als ein mehrere Einzelkriterien überspannendes. In Kapitel 6 geben wir einige Hinweise, wie für ein konkretes Vorgehensmodell ein Kriterienkatalog zum Tailoring entwickelt werden kann.

Die Ergebnisse in Bezug auf Forschungsfrage 2 sind in Tabelle 2.9 zusammengetragen. Es ist leicht zu erkennen, dass die in der Literatur genannten Maßnah-

2.4 Charakterisierung von Projekten

men zum Tailoring allgemein und unspezifisch bleiben. Viele stellen weniger eine konkrete Maßnahme in Bezug auf Anpassung des Vorgehensmodells dar, sondern haben vielmehr den Charakter einer strategischen Richtungsentscheidung. Für den Zusammenhang zwischen Projekteigenschaften und den daraus folgenden Maßnahmen für das Tailoring sei auf den Kriterienkatalog in Anhang A und auf [KK13] verwiesen. In [KK13] ist der Kriterienkatalog in etwas kompakterer Form und die Beziehung zu den Maßnahmen aus Tabelle 2.9 tabellarisch dargestellt.

<i>Maßnahmen in Bezug auf Stakeholder</i>
Kunden stärker einbeziehen Endnutzer stärker einbeziehen Management einbeziehen Trainings für Endnutzer durchführen
<i>Maßnahmen in Bezug auf Projektlebenszyklus</i>
Besonderes Gewicht auf die Anforderungsanalyse legen Besonderes Gewicht auf die Systemarchitektur legen Besonderes Gewicht auf Integration und Test legen Besonderes Gewicht auf das kaufmännische Projektmanagement legen Besonderes Gewicht auf Prototypenentwicklung legen Für kurze Feedback-Zyklen sorgen, beispielsweise durch regelmäßige Lieferungen, On-Site-Customer, usw. Besondere Planungsmuster für zeitkritische Projekte anwenden (z.B. Time Boxing oder Follow-the-Sun)
<i>Maßnahmen in Bezug auf Projektorganisation</i>
Mehr im Projekt dokumentieren, etwa mit Hilfe formalisierter Dokumentvorlagen Weniger im Projekt dokumentieren, etwa durch Ersetzung von Dokumentation durch Stand-Up Meetings Anzahl der Iterationen erhöhen, bzw. Iterationsdauer verkürzen Kommunikationsmuster und -pfade im Projekt formalisieren Offene Kommunikation im Projekt fördern Werkzeuglandschaft im Projekt auf das Gewicht der Vorgehensweise abstimmen
<i>Maßnahmen zum Wissensaufbau und zur Wissenskonservierung</i>
Meetings und Workshops durchführen (Rollenspezifische) Trainings durchführen Technische Infrastruktur für Wissensmanagement bereitstellen

Tabelle 2.9: In der Literatur genannte Maßnahmen zum Tailoring von Vorgehensmodellen

Bezug zur Arbeit

Die vorgestellte systematische Literaturrecherche ist selbst Teil und Beitrag der vorliegenden Arbeit. Wie am Kriterienkatalog in Anhang A zu erkennen, sind mögliche Einflussfaktoren auf eine projektspezifische Ausprägung von Vorgehensmodellen vielfach untersucht. Die daraus resultierenden Maßnahmen zur

konkreten Anpassung eines Vorgehensmodells bleiben jedoch vage und abstrakt (vergleiche Tabelle 2.9). Neben der Tatsache, dass zur Umsetzung einer Maßnahme abhängig von der Organisation, der Vorgehensweise und dem konkreten Projekt interpretiert und ausgestaltet werden muss, mag dieser Umstand damit zusammenhängen, dass die Beschreibung eines Zusammenhangs zwischen Projekteigenschaften und den betroffenen Inhalten in existierenden Vorgehensmodellen nicht oder nur rudimentär möglich ist (vergleiche hierzu insbesondere Kapitel 3). Die vorliegende Arbeit leistet hier einen Beitrag, indem sie durch das feature-basierte Tailoring Vorgehensmodelle dazu befähigt, den Zusammenhang zwischen Projekteigenschaften und Inhalten des Vorgehensmodells anhand klar definierter Regeln überhaupt auszudrücken.

2.5 Verfahren zur Anpassung von Vorgehensmodellen und Methoden

Im folgenden Abschnitt betrachten wir existierende Ansätze zur projektspezifischen Anpassung von Vorgehensmodellen.

Wir gliedern die Betrachtung in die Ansätze des *Situational Method Engineering*, in Ansätze, die Techniken der Produktlinienmodellierung auf Vorgehensmodelle anwenden und in Ansätze, die dem Tailoring nach Definition 3 entsprechen.

2.5.1 Situational Method Engineering

Situational Method Engineering beschreibt die Disziplin der *Methodenkonstruktion* unter Einbeziehung der *Situation* des Projekts. Dabei wird die projektspezifische Vorgehensweise für jedes Projekt neu konstruiert: „For each situation, for instance a project or an organisation, a different method is built“ [Har97].

Ein Überblick über den Prozess des Situational Method Engineering ist in Abbildung 2.16 dargestellt.

Die Schritte im Einzelnen sind:

Charakterisierung der Situation Die Situation beschreibt das konkrete Projekt, für das eine Vorgehensweise konstruiert werden soll. Dieses wird durch sogenannte *Situation Factors* charakterisiert.

Auswahl von Methodenfragmenten Basierend auf dem durch die Situation Factors beschriebenen Profil werden geeignete Methodenfragmente aus einer Methodensammlung (*Method Base*) ausgewählt oder – falls nicht in der Methodensammlung vorhanden – neu erstellt.

Zusammensetzung der Methodenfragmente Die gewählten Methodenfragmente werden zu einem Gesamtverfahren zusammengestellt.

Evaluation der Leistungsfähigkeit Die Leistungsfähigkeit des konstruierten Vorgehens wird überprüft, indem es am „Abschneiden“ des anwendenden Projekts gemessen wird.

Die projektspezifische Erstellung von Vorgehensmodellen basierend auf dem Eclipse Process Framework folgt dem Ansatz des Situational Method Engineering (vergleiche auch Abschnitt 3.2.2). Das Eclipse Process Framework unterstützt die Methodenkonstruktion insbesondere durch die Bereitstellung eines (Meta-)Modells für Methodenfragmente, für die Methodensammlung und die Zusammenstellung der Fragmente mit Hilfe einer Konfiguration. Des Weiteren

2.5 Verfahren zur Anpassung von Vorgehensmodellen und Methoden

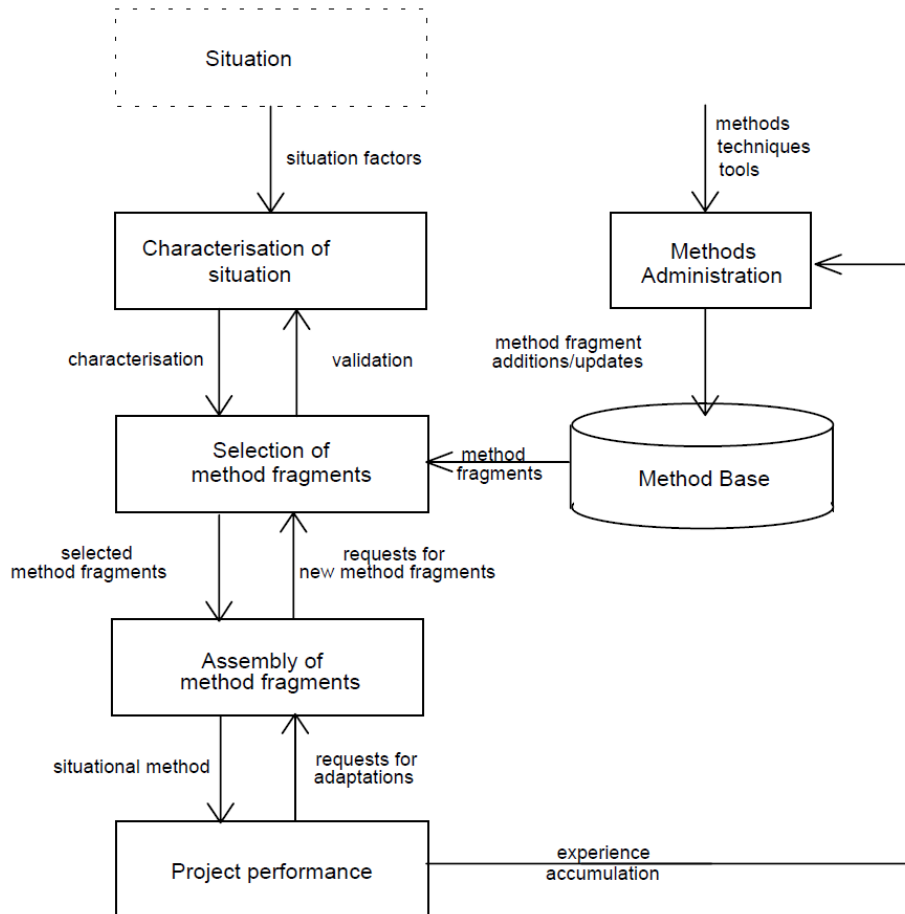


Abbildung 2.16: Prozess des Situational Method Engineering (übernommen aus [Har97])

bietet das Eclipse Process Framework ein komplexes Werkzeug basierend auf der Eclipse Plattform zur Erstellung, Bearbeitung und Verwaltung von Methodenfragmenten, -Sammlungen und Konfigurationen.

Bezug zur Arbeit

Wir machen mit der vorliegenden Arbeit keine Aussage über die Konstruktion von Methoden. Wir nehmen die Inhalte eines Vorgehensmodells (bzw. die Inhalte der Methodensammlung – in Abbildung 2.16 rechts dargestellt) als gegeben an – beispielsweise basierend auf einem Vorgehensstandard. Das Situational Method Engineering stellt deshalb ein wichtiges Verfahren dar, wenn es um die Erstellung projektspezifisch anpassbarer Inhalte für ein Vorgehensmodell geht. Wir greifen die Erstellung von anpassbaren Inhalten in Kapitel 6 wieder auf.

Das Eclipse Process Framework ist die bekannteste Umsetzung des Situational Method Engineering Ansatzes zur Konstruktion projektspezifischer Methoden. Während in [Har97] zwar ein Katalog von Situationsfaktoren vorgeschlagen wird, bleiben die Situationsfaktoren in der Umsetzung unberücksichtigt. Die Charakterisierung der Situation – also des Projekts – ist mit den Mitteln des Eclipse Process Framework nicht möglich. Ebenso bleibt unklar, wie abhängig von einem – mit welchen Situationsfaktoren auch immer beschriebenen – Profil eines Projekts die Methodenfragmente ausgewählt und zusammengestellt werden sollen. Durch die Modellierung der Situationsfaktoren im feature-basierten Tailoring leisten wir in diesem Bereich einen Beitrag und stellen den Zusammenhang zwischen Projektsituation und resultierender Vorgehensweise auf Modellenebene her.

In Bezug auf die Wahl geeigneter Situationsfaktoren gibt [Har97] zu bedenken: „To yield a consistent project characterisation, dependencies between situation factors have to be taken into account“. Als Lösung des Problems der Abhängigkeit zwischen Situationsfaktoren wird in [Har97] die Beschränkung auf einen kleinen Katalog empfohlen. Wäre der Katalog zu groß, würde zudem die Auswahl geeigneter Methoden aus der Methodensammlung zu komplex. Im feature-basierten Tailoring können Abhängigkeiten zwischen Situationsfaktoren direkt im Modell dargestellt werden und bei der „Wahl“ der Vorgehensmodellinhalte entsprechend berücksichtigt werden. Zudem erlaubt die hierarchische Organisation der Situationsfaktoren die Charakterisierung des Projekts durch einen „größeren“ Katalog an Situationsfaktoren.

Situational Method Engineering und Tailoring: Schließt man sich der in Abbildung 2.17 dargestellten Sichtweise von Henderson-Sellers und Ralyte [HSR10] an⁶, so kann zwischen *konstruierten*, *fixen* und *„tailored“* Vorgehensweisen unterschieden werden.

Fixe Vorgehensweisen sind aus den in Kapitel 1 genannten Gründen in der Praxis kaum einsetzbar. Bleiben die projektspezifischen Vorgehen, die aus der Konstruktion oder aus dem projektspezifischen Tailoring eines vorher definierten Vorgehensmodells hervorgehen. Situational Method Engineering beschäftigt sich mit dem ersten Ansatz, während Tailoring die Mechanismen zur Realisierung des zweiten Ansatzes beschreibt.

⁶ Das Plus im Kreis und das Epsilon im Kreis in Abbildung 2.17 repräsentieren die OML Symbole [HSF92] für eine konfigurative bzw. nicht-konfigurative Ganzes-Teil-Beziehung.

2.5 Verfahren zur Anpassung von Vorgehensmodellen und Methoden

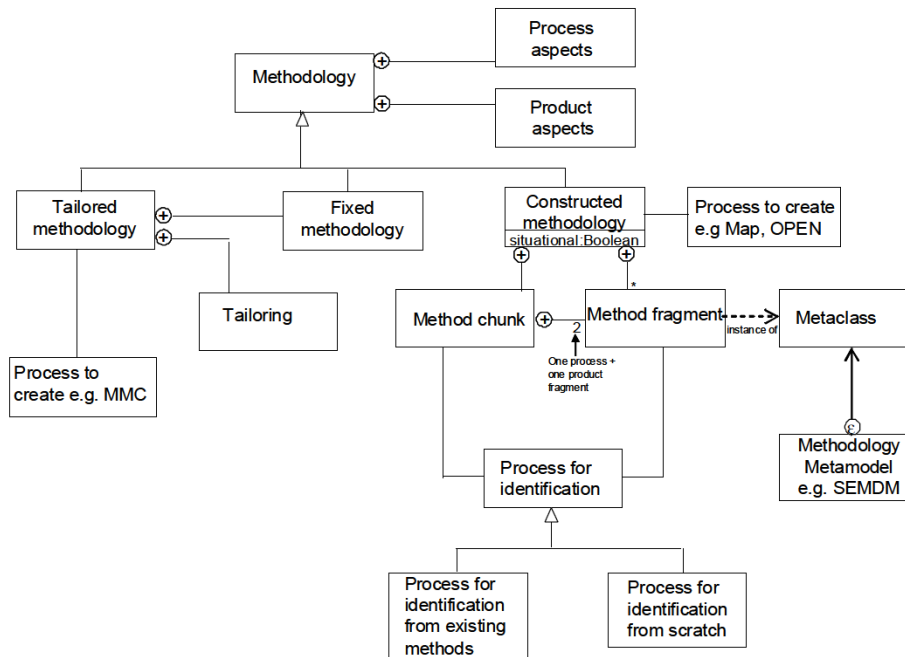


Abbildung 2.17: Modell des Situational Method Engineering Ansatzes (übernommen aus [HSR10])

2.5.2 Prozesslinienmodellierung

Inspiziert vom Software Product Line (SPL) Engineering [SEI11] regt Rombach in [Rom05] die Anwendung der dortigen Konzepte auf Vorgehensmodelle, und damit die Entwicklung von *Prozesslinien*, an. Prozesslinien würden es erlauben, Gemeinsamkeiten („Commonalities“) und Unterschiede („Variabilities“) verschiedener Ausprägungen des Vorgehensmodells zu erfassen. Aus diesen ließe sich je nach Bedarf des vorliegenden Projekts eine konkrete Variante ableiten. Hinsichtlich der Typisierung von [Chr92] (vgl. Abschnitt 2.2.2) schlägt Rombach ein stabiles Kern-Modell (Commonalities) mit ebenfalls vorher definierten und beschriebenen Erweiterungen (Variabilities) vor.

Diese Idee hat mittlerweile in einigen Arbeiten Anwendung gefunden, die wir im Folgenden diskutieren.

In [Ter10, Ter09] wird ein Ansatz zur Modellierung von Entwicklungsmodell-Linien (Development Model Line – kurz: DML) vorgestellt. Die Modellierung nach dem DML Ansatz erlaubt dem Prozessingenieur, die Variabilität im Vorgehensmodell, namentlich konfigurative, erweiternde und modifizierende Variabilität darzustellen.

Das Vorgehensmodell ist in seinem Lebenszyklus allen drei Arten von Variabilität unterworfen: bei der Erstellung und der organisationsspezifischen Anpassung wird es erweitert und modifiziert und bei der projektspezifischen Anpassung konfiguriert.

Insbesondere bei der Beschreibung der konfigurativen Variabilität orientiert sich Ternité in [Ter10] an der Terminologie der Produktlinienmodellierung: wie in Feature-Modellen wird unterschieden zwischen verpflichtenden und optionalen Eigenschaften des Vorgehensmodells. Diese Eigenschaften bestimmen diejenigen Anteile, die bei Konfiguration einer Variante ein- oder ausgeschlossen werden sollen.

Der DML Ansatz aus [Ter10] ist in die Entwicklung des V-Modell XT 1.3 eingeflossen.

In [Kuh07] wird ein integrierter Modellierungsansatz für Vorgehensmodelle (IMV) entwickelt, welcher es erlaubt, *Prozesslinien* zu erstellen und zu pflegen. Motivierender Hintergrund dieser Arbeit waren die Erfahrungen mit verschiedenen Varianten des V-Modell XT. Wesentliche Elemente des integrierten Modellierungsansatzes für Vorgehensmodelle sind:

Modularisierung Stabile und variable Anteile können getrennt voneinander verwaltet und gepflegt werden. Damit können insbesondere Varianten des Vorgehensmodells von Weiterentwicklungen des zentral gepflegten Kerns profitieren.

Beziehungen als Modellelemente Beziehungen zwischen Modellelementen (z. B. zwischen Produkten, Aktivitäten und Rollen) sind selbst vollwertige Modellelemente. Damit lassen sich Beziehungen variieren, wie andere Modellelemente auch.

Wesentliche Konzepte aus [Kuh07] sind in die Version 1.3 des V-Modell XT eingegangen.

Bezug zur Arbeit

Der IMV Ansatz dient im Kern der organisationspezifischen Variantenbildung von Vorgehensmodellen. Dabei stützt er sich auf die aus der Produktlinienmodellierung bekannte Trennung von statischen und variablen Anteilen des Vorgehensmodells ab. Das Tailoring – also die Anpassung einer im Sinne des IMV Ansatzes „fertigen“ Variante – liegt außerhalb der Betrachtung in [Kuh07]. Die Bildung organisationspezifischer Varianten liegt im Lebenszyklus eines Vorgehensmodells *vor* der projektspezifischen Anpassung und damit dem Thema der vorliegenden Arbeit.

Der DML Ansatz streift zwar die konfigurative Variation eines Vorgehensmodells bei der projektspezifischen Anpassung und bedient sich in der Terminologie bei der Feature-Modellierung. Allerdings bleibt die Semantik von Feature-Modellen ungenutzt. Mit dem Ansatz des feature-basierten Tailorings knüpfen wir an diese Arbeit an und machen uns die Techniken der Feature-Modellierung zunutze, um eine ausdrucksmächtige Beschreibung der Projektcharakteristika und der Variationspunkte beim Tailoring zu entwickeln.

Der in der vorliegenden Arbeit entwickelte Ansatz zum feature-basierten Tailoring kann aufgefasst werden als Beitrag zu den in [Rom05] formulierten Forschungsaufgaben: „The design of process modeling languages with features for variability specification“ und „More effective methods for creating empirically grounded process/product models“.

2.5.3 Tailoring-Ansätze

Der Definition 3 für Tailoring folgend, wird bei Tailoring-Ansätzen ein gegebenes Vorgehensmodell mit festgelegten Variationspunkten und Kriterien zur Charakterisierung des Projekts angenommen. Der Variationsraum möglicher projektspezifischer Vorgehensmodelle ergibt sich aus der Kombinatorik aller Projektcharakteristika.

Bekannt, diesem Anpassungsansatz folgende, Vorgehensmodelle sind das V-Modell XT (siehe Abschnitt 3.2.1), in Teilen die schweizerische Projektführungs-

2.5 Verfahren zur Anpassung von Vorgehensmodellen und Methoden

methode HERMES (siehe Abschnitt 3.2.3) und das SE Book der T-Systems International GmbH (siehe Abschnitt 3.2.4). Für eine detaillierte Analyse der Anpassungsverfahren dieser Vorgehensmodelle sei auf Kapitel 3 verwiesen.

Bezug zur Arbeit

Die hier beschriebenen Tailoring-Ansätze beinhalten ein rudimentäres Modell zur Beschreibung der Projektcharakteristika. Dabei ist die Menge der Projektcharakteristika und die Beziehungen zwischen diesen durch das Modell vorgegeben (wie im V-Modell XT) oder Querbeziehungen zwischen Charakteristika sind überhaupt nicht darstellbar (wie im SE Book). Der Tailoring-Ansatz in HERMES erlaubt nur die Wahl einer vorkonfigurierten Variante der Vorgehensweise.

Der in dieser Arbeit entwickelte Ansatz des feature-basierten Tailorings fällt in diese Klasse der Anpassungsverfahren. Gegenüber existierenden Ansätzen umfasst er ein mächtiges Modell zur Beschreibung der Projektsituation, der Querbeziehungen zwischen Projektcharakteristika und der Wirkung der Projektcharakteristika auf die Inhalte einer projektspezifischen Ausprägung des Vorgehensmodells.

2.5.4 Projektspezifische Anpassung agiler Vorgehensweisen

Die projektspezifische Anpassung agiler Vorgehensweisen findet in der Regel kontinuierlich während des Projekts statt, indem z. B. zum Ende jeder Iteration über das Vorgehen im Projekt reflektiert wird. In der Beschreibung zu Crystal Clear [Coc04] finden sich einige Techniken zur Reflexion und zur Definition einer Methode zu Projektbeginn. Neben Interviews von Mitgliedern anderer Teams zur Sammlung von Erfahrungen besteht die Technik im Kern in der Durchführung von Workshops zum *Methodology Shaping* und zur *Reflection*, bei denen in Bezug auf die Methode folgende Fragen beantwortet werden sollen: Was hat funktioniert? Was hat nicht funktioniert? Welche Methoden oder Techniken sollen in der nächsten Iteration ausprobiert werden (also in die Gesamtmethode aufgenommen werden)?

Bezug zur Arbeit

Wie agile Vorgehensweisen insgesamt eignen sich derartige Verfahren zur Methodenanpassung in erster Linie für kleine Teams. Die erwähnte Methode Crystal Clear ist z. B. für ein Team mit 3 - 6 Mitgliedern vorgesehen. Zudem eignen sich Reflexions-Workshops nur zur Verfeinerung der Praktiken im Projekt. Angepasste Dokumentvorlagen, eine Vorlage für den Projektplan usw. entstehen daraus nicht.

Das feature-basierte Tailoring befasst sich mit der werkzeuggestützten projektspezifischen Anpassung eines reichhaltigen Vorgehensmodells. Im Mikrokosmos des Teams sollte trotz eines projektspezifisch angepassten Vorgehensmodells kontinuierlich über Praktiken im Team reflektiert werden. Die Techniken der verschiedenen agilen Vorgehensweisen stellen hierfür eine gute Vorlage dar.

2.6 Feature-Oriented Domain Analysis

In diesem Abschnitt geben wir einen kurzen Überblick über relevante Erkenntnisse aus dem Bereich der *Feature-Oriented Domain Analysis*. Wir beschreiben *Feature-Modelle* und stellen eine im Laufe dieser Arbeit verwendete Notation für diese Modelle dar. Weiterhin fassen wir kurz zusammen, wie *Feature-Modelle* mit Hilfe der Aussagenlogik und SAT Solvern analysiert werden können.

2.6.1 Feature-Modelle

Feature-Modelle [KCH⁺90, CE00] dienen in der Produktlinienmodellierung der kompakten Darstellung aller Produkte einer Produktlinie. Ein Feature-Modell beschreibt alle variablen Anteile an einer Produktlinie und deren Abhängigkeiten untereinander. Die Produkte einer Produktlinie werden anhand ihrer *Features* unterschieden. Ein *Feature* ist ein vom Benutzer des Produkts wahrnehmbares und relevantes, wesentliches, unterscheidendes Merkmal, Charakteristikum oder eine Qualität des Produkts.

Ein Feature kann als binärer Schalter aufgefasst werden: ist das Feature gewählt, so ist die durch das Feature beschriebene Eigenschaft des Domänenkonzepts Teil des resultierenden Produkts. Ist das Feature nicht gewählt, so hat das resultierende Produkt die Eigenschaft nicht. Die An- und Abwahl einzelner Features unterliegt dabei den durch das Feature-Modell beschriebenen Regeln.

Im Gegensatz zur Verwendung des Begriffs *Produkt* im Zusammenhang mit Vorgehensmodellen verstehen wir im Zusammenhang mit Produktlinienmodellierung unter einem *Produkt* folgendes: ein *Produkt* ist ein konkretes, durch die jeweilige An- oder Abwahl aller Features entstehendes Mitglied der Produktlinie. *Gültige Produkte* sind diejenigen Produkte, deren Belegung von Features gültig im Sinne des Feature-Modells ist.

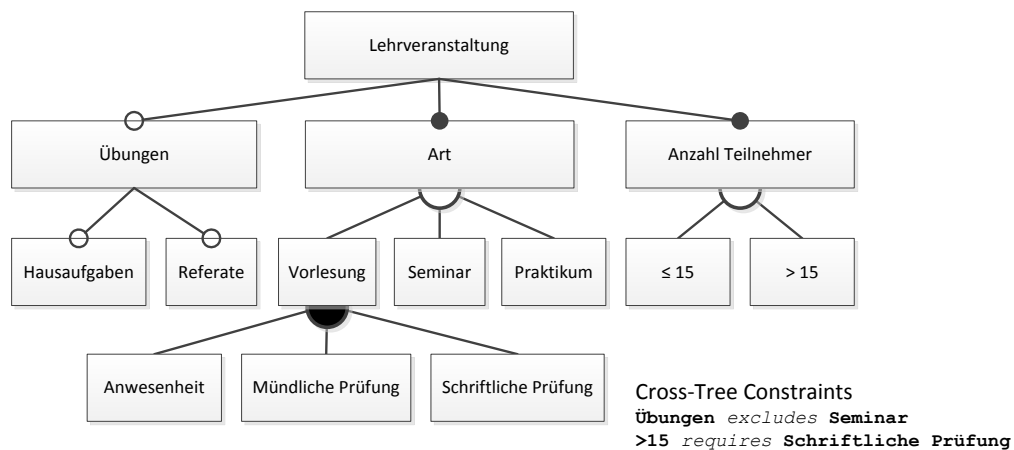


Abbildung 2.18: Feature-Modell Beispiel

Abbildung 2.18 zeigt ein einfaches Feature-Modell einer Lehrveranstaltung. Es besteht aus einem Feature-Baum und einer Reihe quer dazu liegender *Cross-Tree Constraints* (in Abbildung 2.18 rechts unten eingezeichnet). Im Feature-Baum stellen die Knoten die *Features* dar. Die Kanten beschreiben die Beziehungen zwischen Features. Die Wurzel des Baums beschreibt das zu beschreibende Domänenkonzept – im Beispiel der Abbildung 2.18 die Lehrveranstaltung.

2.6 Feature-Oriented Domain Analysis

Optionale Features werden durch einen nicht ausgefüllten Kreis dargestellt (in der Abbildung z.B. *Übungen*). Diese können nur Bestandteil eines Produkts sein, wenn das Eltern-Feature auch Teil des Produkts ist. Beispielsweise können *Hausaufgaben* nur Bestandteil der *Lehrveranstaltung* sein, wenn diese auch *Übungen* beinhaltet.

Verpflichtende Features, z. B. *Anzahl Teilnehmer*, werden durch einen ausgefüllten Kreis gekennzeichnet. Diese sind Bestandteil aller Produkte, die auch das Eltern-Feature einschließen.

Features können weiterhin zu Gruppen zusammengefasst werden, um die Beziehung der Features untereinander zu beschreiben. Die Features *Anwesenheit*, *mündliche Prüfung* und *Schriftliche Prüfung* stehen beispielsweise in einer OR-Beziehung zueinander (symbolisiert durch einen ausgefüllten Halbkreis am Eltern-Feature) – d.h. mindestens eines dieser Features muss gewählt sein, wenn das Eltern-Feature gewählt ist. Die Features *Vorlesung*, *Seminar* und *Praktikum* wiederum stehen in einer XOR-Beziehung (gekennzeichnet durch den nicht ausgefüllten Halbkreis am Eltern-Feature). Ist das Eltern-Feature der Gruppe Bestandteil des Produkts, so ist *genau eins* der Kinder in der Gruppe zu wählen.

Zusätzliche Abhängigkeiten zwischen Features können über *Cross-Tree Constraints* ausgedrückt werden. Für die vorliegende Arbeit beschränken wir uns auf *requires* und *excludes* Constraints⁷. Ein Constraint $A \text{ requires } B$ ist semantisch äquivalent zum aussagenlogischen Ausdruck $A \rightarrow B$. Ein Constraint $A \text{ excludes } B$ ist semantisch äquivalent zum Ausdruck $\neg A \vee \neg B$. Das Feature-Modell in Abbildung 2.18 beinhaltet zwei Cross-Tree Constraints (rechts unten in der Abbildung). Der erste zwischen > 15 und *Schriftliche Prüfung* besagt: „Ist die Anzahl der Teilnehmer an der Lehrveranstaltung größer als 15, so ist eine schriftliche Prüfung durchzuführen“. Mit anderen Worten: das Feature > 15 impliziert das Feature *Schriftliche Prüfung*. Der zweite Cross-Tree Constraint besagt: die Wahl des Features *Seminar* schließt die Wahl des Features *Übungen* aus.

Die Abhängigkeiten im Feature-Modell bestehen aus der Konjunktion der durch den Baum definierten Beziehungen und der Menge der Cross-Tree Constraints. In der Literatur [CE00] finden sich zahlreiche alternative Notationen für Feature-Modelle. Vielfach werden beispielsweise zu Featuregruppen *Kardinalitäten* angegeben. Durch $[n, m]$ Featuregruppen ließe sich angeben, dass mindestens n und höchstens m Kind-Features aus der Gruppe gewählt werden dürfen (m ist dabei immer kleiner gleich der Anzahl Features in der Gruppe).

Im feature-basierten Tailoring verwenden wir Kardinalitäten zur Definition von Featuregruppen nicht. Wir verwenden lediglich die durch *Oder* und *exklusives Oder* verknüpften Gruppen. In „Kardinalitätsschreibweise“ entspricht die erste einer $[1, m]$ Gruppe (mit m kleiner gleich der Anzahl Features in der Gruppe) und die zweite einer $[1, 1]$ Gruppe.

Ein gültiges Produkt im Sinne eines Feature-Modells lässt sich darstellen als die Menge der im Produkt eingeschlossenen Features. Im Beispiel aus Abbildung 2.18 wäre $L_1 = \{\text{Lehrveranstaltung, Art, Anzahl Teilnehmer, Vorlesung, } > 15, \text{ Schriftliche Prüfung}\}$ ein gültiges Produkt, während $L_2 = \{\text{Lehrveranstaltung, Art, Anzahl Teilnehmer, Vorlesung, } > 15\}$ kein gültiges Produkt wäre. Im Fall von L_2 ist die Bedingung verletzt, dass bei Wahl des Features *Vorlesung* als Prüfungskriterium mindestens ein Feature aus der Menge $\{\text{Anwesenheit, mündliche Prüfung, Schriftliche Prüfung}\}$ zu wählen ist. Weiterhin ist durch einen Cross-

⁷ In manchen Varianten der Feature-Modellierung werden beliebige aussagenlogische Ausdrücke zwischen Features zugelassen, siehe etwa [Bat05, MBC09, KTS⁺09].

Tree Constraint gefordert, dass die Wahl des Features > 15 die Wahl des Features *Schriftliche Prüfung* bedingt. Auch diese Bedingung ist im Fall von L_2 verletzt.

2.6.2 Analyse von Feature-Modellen

Bei der Analyse von Feature-Modellen kann unterschieden werden zwischen der Überprüfung der Korrektheit (also ob das Modell Features beinhaltet, die an keinem Produkt teilnehmen, also *tot* sind oder ob es überhaupt ein gültiges Produkt gibt, usw.) und der Unterstützung bei der interaktiven Konfiguration. Beide Bereiche spielen bei den entwickelten Verfahren zum feature-basierten Tailoring eine wesentliche Rolle.

Korrektheit

Laut [MWC09] besteht die Überprüfung der Korrektheit eines Feature-Modells beispielsweise in der *Konsistenzprüfung*, der Identifikation *toter* oder *gemeinsamer* Features und der Äquivalenzprüfung.

Ein Feature-Modell wird in [MWC09] als *inkonsistent* bezeichnet, wenn es keine Belegung gibt, welche ein Produkt erzeugt. Im linken Teil der Abbildung 2.19 ist ein einfaches Beispiel für ein derartiges Feature-Modell dargestellt. Die Features B und C sind beide verpflichtende Kinder von A , d.h. bei Wahl von A müssen sie beide gewählt werden. Ein Cross-Tree Constraint besagt wiederum, dass die Features B und C sich gegenseitig ausschließen. Insgesamt ist es damit nicht möglich, ein gültiges Produkt zu konfigurieren.

Tote Features nennen [MWC09] solche Features, die in keinem gültigen Produkt teilnehmen. Der mittlere Teil von Abbildung 2.19 veranschaulicht so einen Fall: die Features B und C sind alternativ zu wählen. Ein Cross-Tree Constraint besagt allerdings, dass bei Wahl von C gleichzeitig B zu wählen ist. Die Wahl von C führt damit nie zu einem gültigen Produkt. Das einzig gültige Produkt zum Feature-Modell im mittleren Teil von Abbildung 2.19 ist $P = \{A, B\}$. B ist gleichzeitig ein *gemeinsames* Feature. Es nimmt an allen gültigen Produkten des Feature-Modells teil. Abgesehen von der Wurzel können gemeinsame Features ein Hinweis auf eine „ungünstige“ Modellierung der Variabilität hinweisen: da diese Features an jedem gültigen Produkt teilnehmen, stellen sie keinen variablen Anteil dar.

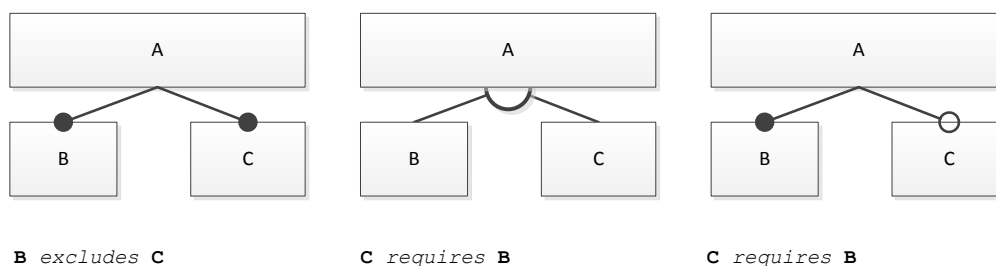


Abbildung 2.19: Feature-Modell Korrektheit

Neben den hier vorgestellten gibt es zahlreiche weitere Korrektheitskriterien für Feature-Modelle, die häufig zumindest auf Optimierungspotenzial bei der Modellierung der Variabilität hinweisen. Einen Überblick über derartige Eigenschaften von Feature-Modellen bietet [vdML04]. Dort wird etwa *Redundanzfrei-*

2.6 Feature-Oriented Domain Analysis

heit als eine wünschenswerte Eigenschaft von Feature-Modellen genannt. Im rechten Teil von Abbildung 2.19 ist der Cross-Tree Constraint redundant, da das Feature B bei Wahl von A sowieso verpflichtend zu wählen ist. Wir haben die in [vdML04] zusammengefassten Eigenschaften im Modellierungswerkzeug für das Feature-Modell des SE Book prototypisch implementiert (siehe Abschnitt 7.2).

Äquivalenzprüfung von Feature-Modellen ist besonders bei der Bearbeitung des Modells hilfreich: hiermit lässt sich prüfen, ob zwei Feature-Modelle die gleiche Produktmenge haben. Betreibt man beispielsweise *Refactoring* eines Feature-Modells, ist man üblicherweise daran interessiert, dass das Modell vorher und nachher dieselbe Produktmenge ergibt, auch wenn sich die Modelle strukturell unterscheiden.

Interaktive Konfiguration

Die Verfahren zur Unterstützung der interaktiven Konfiguration stellen sicher, dass bei der Konfiguration eines Feature-Modells die darin festgeschriebenen Bedingungen eingehalten werden und damit nur *gültige Produkte* im Sinne des Feature-Modells erstellt werden können.

Feature-Modelle als aussagenlogischer Ausdruck

Die automatisierte Analyse von Feature-Modellen wird ermöglicht durch die Tatsache, dass sich Feature-Modelle mit ein paar einfachen Regeln in einen aussagenlogischen Ausdruck übersetzen lassen [Bat05, CW07]. Damit lassen sich Schlussfolgerungen über das Modell ziehen – beispielsweise durch den Einsatz von Satisfiability (SAT) Solvers. Tabelle 2.10 zeigt die Regeln zur Überführung eines Feature-Modells in einen aussagenlogischen Ausdruck [Bat05, CW07].

Nr.	Beziehung im Feature-Modell	entsprechender Ausdruck
(1)	F ist das Wurzelfeature	F
(2)	F ist Elternfeature von K	$K \Rightarrow F$
(3)	K ist verpflichtendes Kind von F	$F \Rightarrow K$
(4)	F ist Elternfeature von $OR(K_1, \dots, K_n)$	$F \Rightarrow (K_1 \vee \dots \vee K_n)$
(5)	F ist Elternfeature von $XOR(K_1, \dots, K_n)$	$\forall (K_1, K_2) : K_1 \Rightarrow \neg K_2$
(6)	Cross-Tree Constraint	lässt sich direkt als aussagenlogischer Ausdruck formulieren

Tabelle 2.10: Feature-Modelle als aussagenlogischer Ausdruck

Jedes Feature entspricht einer Variablen im Ausdruck während der Feature-Baum und die Cross-Tree Constraints die logischen Beziehungen festlegen. Da die Wurzel W Teil jedes gültigen Produkts ist, wird sie durch den einfachen Ausdruck (W) repräsentiert. Ein Kind-Feature kann nur dann gewählt werden, wenn auch das Elternfeature gewählt ist. Deshalb gibt es eine Implikation von jedem Kind zu seinem Elternfeature. Ist ein Feature ein verpflichtendes Kind, gibt es weiterhin eine Implikation vom Elternfeature zum Kind. *OR* und *XOR* Gruppen werden durch eine Implikation vom Elternfeature zu der entsprechend

verknüpften Menge von Kindern dargestellt. Um einen das gesamte Feature-Modell darstellenden Ausdruck zu erhalten, werden die Teilausdrücke (inklusive der Ausdrücke für die Cross-Tree Constraints) konjugiert.

Wendet man die Regeln aus Tabelle 2.10 auf die Feature-Modelle aus Abbildung 2.19 an, so ergeben sich folgende Ausdrücke (unterhalb der Teilausdrücke ist jeweils die Nummer der angewendeten Regel aus Tabelle 2.10 angegeben):

$$M_1 = \underset{(1)}{A} \wedge \underset{(2)}{B} \Rightarrow \underset{(2)}{A} \wedge \underset{(2)}{C} \Rightarrow \underset{(2)}{A} \wedge \underset{(3)}{A} \Rightarrow \underset{(3)}{B} \wedge \underset{(3)}{A} \Rightarrow \underset{(3)}{C} \wedge \underset{(6)}{B} \Rightarrow \underset{(6)}{\neg C}$$

$$M_2 = \underset{(1)}{A} \wedge \underset{(2)}{B} \Rightarrow \underset{(2)}{A} \wedge \underset{(2)}{C} \Rightarrow \underset{(2)}{A} \wedge \underset{(5)}{B} \Rightarrow \underset{(5)}{\neg C} \wedge \underset{(5)}{C} \Rightarrow \underset{(5)}{\neg B} \wedge \underset{(6)}{C} \Rightarrow \underset{(6)}{B}$$

$$M_3 = \underset{(1)}{A} \wedge \underset{(2)}{B} \Rightarrow \underset{(2)}{A} \wedge \underset{(2)}{C} \Rightarrow \underset{(2)}{A} \wedge \underset{(3)}{A} \Rightarrow \underset{(3)}{B} \wedge \underset{(6)}{C} \Rightarrow \underset{(6)}{B}$$

Am Ausdruck M_2 ist deutlich zu erkennen, dass das Feature C im mittleren Feature-Modell Beispiel aus Abbildung 2.19 an keiner gültigen Konfiguration teilnehmen kann.

Feature-Modelle als Erfüllbarkeitsproblem

Werden Feature-Modelle als aussagenlogischer Ausdruck aufgefasst, können diese als Erfüllbarkeitsproblem der Aussagenlogik analysiert werden. Es untersucht, ob eine gegebene aussagenlogische Formel *erfüllbar* ist, d.h. ob eine Variablenbelegung existiert, für die die gegebene Formel als *wahr* ausgewertet werden kann. Die Formel $(A \wedge B)$ ist beispielsweise *wahr* für die Variablenbelegung $A = \text{wahr}, B = \text{wahr}$. Die Formel $(A \wedge B \wedge \neg A)$ hingegen ist nicht erfüllbar: es gibt keine Belegung von A , für die die Formel *wahr* wird.

Zur Entscheidung von Erfüllbarkeitsproblemen der Aussagenlogik kommen üblicherweise SAT-Solver zum Einsatz (SAT vom englischen „satisfiability“). Damit ein aussagenlogischer Ausdruck von einem SAT-Solver verarbeitet werden kann, muss dieser üblicherweise in der *konjunktiven Normalform* (CNF) vorliegen. Eine aussagenlogische Formel ist dann in konjunktiver Normalform, wenn sie aus einer Konjunktion von Disjunktionstermen besteht, d.h. jeder der konjugierten Terme ist eine Disjunktion von Literalen.

Durch die in Tabelle 2.11 dargestellten Vorschriften lässt sich ein aussagenlogischer Ausdruck in die konjunktive Normalform überführen.

Satisfiability (SAT) Solver

SAT-Solver sind Verfahren zur Entscheidung eines aussagenlogischen Erfüllbarkeitsproblems. In der vorliegenden Arbeit verwenden wir SAT-Solver zur Entscheidung, ob die Wahl bestimmter Projektcharakteristika gültig im Sinne der durch das Konfigurationsmodell vorgeschriebenen Regeln ist. Die Regeln werden dabei beschrieben durch ein Feature-Modell – die Projektcharakteristika entsprechen den Features (bzw. ihrer Kombination) im Baum. Die Wahl von Projektcharakteristika ist demnach genau dann gültig im Sinne des Tailorings,

2.6 Feature-Oriented Domain Analysis

Nr.	Ausdruck	konjunktive Normalform
(1)	$A \Rightarrow B$	$\neg A \vee B$
(2)	$A \Leftarrow B$	$A \vee \neg B$
(3)	$A \Leftrightarrow B$	$(\neg A \vee B) \wedge (A \vee \neg B)$
(4)	$\neg\neg A$	A
(5)	$\neg(A \wedge B)$	$\neg A \vee \neg B$
(6)	$\neg(A \vee B)$	$\neg A \wedge \neg B$
(7)	$A \vee (B \wedge C)$	$(A \vee B) \wedge (A \vee C)$

Tabelle 2.11: Aussagenlogischer Ausdruck in konjunktiver Normalform

wenn die Wertebelegung eine Lösung für den dem Feature-Baum entsprechenden aussagenlogischen Ausdruck ist.

Im Rahmen der vorliegenden Arbeit zum Einsatz gekommene SAT-Solver sind die Bibliothek Sat4J [BP10], eine Java-Implementierung des MiniSat-Verfahrens [ES03] und die Microsoft Solver Foundation [Mic12a].

Bezug zur Arbeit

Wir verwenden die Techniken der Feature-Modelle im feature-basierten Tailoring zur Beschreibung der Projektcharakteristika und ihrer Abhängigkeiten. Damit können wir von den Erkenntnissen der entsprechenden Forschung profitieren und diese für die projektspezifische Anpassung von Vorgehensmodellen nutzen.

Der Prozessingenieur wird unterstützt durch die automatisierte Überprüfung der Korrektheit des Konfigurationsmodells – beispielsweise in einem Modellierungswerkzeug. Wie in Kapitel 3 angedeutet, ist der Prozessingenieur bei der Beschreibung der variablen Anteile und der Konfigurationsmöglichkeiten in aktuellen Vorgehensmodellen zu großen Teilen auf sein Fachwissen angewiesen. Ungültige Konfigurationsmöglichkeiten werden oft erst bei der Durchführung der Konfiguration selbst festgestellt. In der Umsetzung des feature-basierten Tailorings für das SE Book haben wir ein prototypisches Werkzeug zur Überprüfung der Korrektheit des Konfigurationsmodells entwickelt (siehe Abschnitt 7.2).

Die Unterstützung der interaktiven Konfiguration spielt eine wesentliche Rolle für den Anwender des Vorgehensmodells. Aufgrund der Komplexität moderner Vorgehensmodelle ist es nicht mehr vorstellbar, die projektspezifische Anpassung ohne geeignete Werkzeugunterstützung durchzuführen. Ein Ziel bei der Entwicklung eines Werkzeugs zur Durchführung des Tailorings sollte sein, dass der Anwender nur *gültige Produkte* (sprich: projektspezifische Ausprägungen) im Sinne des Feature-Modells erzeugen kann. Um dies sicherzustellen, eignen sich die Analyseverfahren zur Unterstützung bei der interaktiven Konfiguration. In den Fallstudien zum SE Book und zum V-Modell XT stellen wir für jedes Vorgehensmodell jeweils ein (prototypisches) Werkzeug zur interaktiven Konfiguration vor (vergleiche Abschnitte 7.2 und 7.3).

2.7 Zusammenfassung

Im vorangegangenen Kapitel haben wir einen Überblick über für die vorliegende Arbeit relevante Grundlagen und verwandte Arbeiten gegeben. Wir haben den Begriff des Vorgehensmodells eingegrenzt und das Tailoring im Lebenszyklus eines Vorgehensmodells gegen andere Anpassungsschritte positioniert.

Da die Grenze zwischen Anpassung eines Vorgehensmodells und Wahl einer Variante oder Ausprägung teilweise fließend ist, haben wir Arbeiten zur Klassifizierung und Eignungseinstufung von Vorgehensmodellen betrachtet.

Weiterhin haben wir uns mit Arbeiten befasst, die sich mit der Charakterisierung von Projekten zum Zwecke der projektspezifischen Anpassung eines Vorgehensmodells beschäftigen. Die Ergebnisse dieser Betrachtung sind in den Kriterienkatalog zum Tailoring in Anhang A eingeflossen. In diesen Bereich fällt auch eine zusammen mit Partnern von den Telekom Laboratories und der T-Systems International GmbH durchgeführte Studie zur Ermittlung des Anpassungsbedarfs für das SE Book der T-Systems. Diese wurde in den Kontext vergleichbarer Arbeiten gestellt.

Schließlich haben wir existierende Verfahren zur projektspezifischen Anpassung diskutiert und gegeneinander positioniert. Im folgenden Kapitel 3 werden wir diese Thematik wieder aufgreifen und einige ausgewählte Vorgehensmodelle hinsichtlich ihrer Anpassungsfähigkeit und dem zugrunde liegenden Konzept näher untersuchen.

Den Abschluss dieses Kapitels bildete ein Überblick über für die Arbeit wesentliche Verfahren und Theorien der Feature Oriented Domain Analysis. Wir haben Notation und Semantik von Feature-Modellen dargestellt und wie diese mit den Mitteln der Aussagenlogik ausgewertet werden können.

Anpassungsfähigkeit von Vorgehensmodellen

Nachdem in Kapitel 2 für die vorliegende Arbeit wesentliche Begriffe vereinbart wurden und ein allgemeiner Überblick über den Stand in Forschung und Praxis zur Anpassung von Vorgehensmodellen gegeben wurde, untersuchen wir in diesem Kapitel die Anpassungsfähigkeit einiger ausgewählter Vorgehensmodelle im Detail. Die Analyse der Fähigkeit zur projektspezifischen Anpassung in der Praxis eingesetzter Vorgehensmodelle bildet eine Grundlage und Motivation für die vorliegende Arbeit.

Am Ende dieses Kapitels wird der Leser einen Überblick über die Mächtigkeit und Grenzen der Anpassungsmechanismen in populären Vorgehensmodellen haben.

Übersicht

3.1	Überblick	66
3.2	Reichhaltige Vorgehensmodelle	67
3.3	Leichtgewichtige Vorgehensweisen	81
3.4	Zusammenfassung	84

3.1 Überblick

Wir betrachten in diesem Kapitel erstens die Möglichkeiten und das Maß an Unterstützung für den Anwender bei der Durchführung des Tailorings und zweitens die Variabilitätspunkte, also derjenigen Modellanteile, die potenziell vom Tailoring betroffen sein können.

3.1 Überblick

Abbildung 3.1 illustriert den Aufbau dieses Kapitels: Im ersten Teil betrachten wir die Anpassungsfähigkeit reichhaltiger Vorgehensmodelle. Stellvertretend betrachten wir hier die Mechanismen zur projektspezifischen Anpassung im V-Modell XT (Abschnitt 3.2.1), in auf dem Eclipse Process Framework basierenden Vorgehensmodellen (Abschnitt 3.2.2) und dort speziell bei HERMES (Abschnitt 3.2.3) und schließlich im SE Book – dem hausinternen Vorgehensmodell der T-Systems International GmbH (Abschnitt 3.2.4).

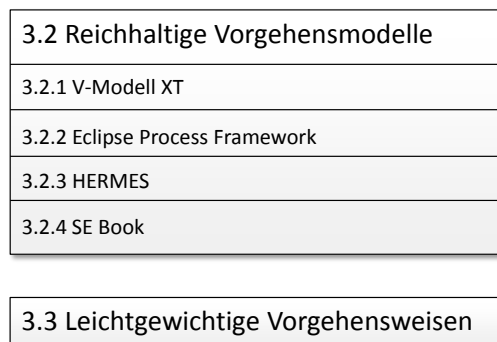


Abbildung 3.1: Struktur des Kapitels 3

Im zweiten Teil untersuchen wir gemeinhin als *leichtgewichtig* (oder im Englischen auch „low-ceremony methodology“ [Coc06]) eingestufte Vorgehensweisen. Das *Gewicht* einer Vorgehensweise wird bei [Coc06] als das Produkt aus folgenden Faktoren beschrieben:

Size Die Anzahl der beschriebenen Elemente. Jedes Arbeitsergebnis, jeder Standard, jede Aktivität, Beschreibungstechnik, usw. zählen als Elemente der Vorgehensweise.

Amount of Ceremony Das Maß für die Präzision oder Toleranz in den Vorgaben. Zur Illustration dieses Maßes wird in [Coc06] folgendes Beispiel genannt: Ein Team skizziert seine Anwendungsfälle auf Servietten und bespricht sie während des Mittagessens. Ein anderes Team füllt für jeden Anwendungsfall eine dreiseitige Vorlage aus und führt einen halbtägigen Workshop zum Review der Anwendungsfälle durch. Beide Gruppen schreiben und besprechen Anwendungsfälle. Das erste mit einem geringen, das zweite mit einem hohen Maß an „Zeremoniell“.

Bei dem so beschriebenen *Gewicht* einer Vorgehensweise handelt es sich nicht um ein tatsächlich rechnerisch zu bestimmendes Produkt, sondern um ein konzeptuelles Maß, welches bei der Klassifizierung helfen soll.

Nach der Betrachtung der Anpassungsfähigkeit reichhaltiger Vorgehensweisen untersuchen wir im zweiten Teil dieses Kapitels (in Abschnitt 3.3) Vorgehensweisen, denen ein im obigen Sinne geringes Gewicht beigemessen werden kann,

da sie erstens nur wenige Elemente (beispielsweise Arbeitsergebnisse und Rollen) definieren und da sie zweitens in der Beschreibung der Methoden und Praktiken ein geringes Maß an Zeremoniell fordern.

3.2 Reichhaltige Vorgehensmodelle

Reichhaltige Vorgehensmodelle basieren auf einem Metamodell, welches die Strukturen des Vorgehensmodells beschreibt. Artefakte, Aktivitäten, Rollen und Abläufe im Vorgehensmodell sind detailliert und strukturiert beschrieben und über ein Beziehungsgeflecht miteinander verbunden. Unter reichhaltigen Vorgehensmodellen verstehen wir insbesondere solche Modelle, die den in Abschnitt 2.2.1 formulierten Anforderungen genügen.

Die strukturierte Beschreibung eines Vorgehensmodells erlaubt die Verarbeitung durch (Software-)Werkzeuge, beispielsweise zur Erstellung von Vorlagen für die Artefakte oder zur Generierung von (initialen) Projektplänen aus den Ablaufstrukturen.

Reichhaltige, strukturierte Vorgehensmodelle sind in der Regel komplex und umfassen oft hunderte bis tausende Modellelemente. Dieser Umstand macht die Anpassung an projektspezifische Gegebenheiten ein lohnenswertes (und notwendiges) Unterfangen.

3.2.1 Das V-Modell XT

Das V-Modell XT ist ein in Deutschland entwickeltes Vorgehensmodell. Es bildet den Entwicklungsstandard für IT-Projekte der öffentlichen Hand in Deutschland. Es existieren zahlreiche Erweiterungen dieses Modells – beispielsweise das V-Modell XT Bund [Bun10], welches organisatorische Besonderheiten der öffentlichen Verwaltung in Deutschland berücksichtigt oder das V-Modell XT BNetzA, welches Besonderheiten der Bundesnetzagentur berücksichtigt und in dieser Behörde für interne Entwicklungsprojekte eingesetzt wird.

Anpassungsfähigkeit

Bei der Betrachtung der Möglichkeiten und Verfahren zur Anpassung des V-Modell XT [RHB⁺06, KTF09] kann unterschieden werden zwischen den Anpassungen, die ein Prozessingenieur während der Entwicklung des Vorgehensmodells und dessen organisationspezifischer Anpassung vornehmen kann und dem eigentlichen Tailoring, welches zu Projektbeginn durchgeführt wird. Die Trennung zwischen beiden Anpassungsschritten äußert sich insbesondere durch die zum Einsatz kommenden Werkzeuge: die Entwicklung einer organisationspezifischen Variante erfolgt im *V-Modell XT Editor* – das Tailoring zu Projektbeginn erfolgt im *V-Modell XT Projektassistent*.

Erstellung eines organisationspezifischen Vorgehensmodells: Für die Erstellung einer organisationspezifischen Variante¹ lassen sich grob drei Verfahren unterscheiden [KTF09]:

Neuaufbau: Basierend auf dem Metamodell des V-Modell XT werden alle Inhalte des Vorgehensmodells neu erstellt. Eine so erstellte organisationspezifische Variante hat mit dem V-Modell XT bis auf das Fundament des Metamodells und möglicherweise der Nutzung der Werkzeuge nichts gemein.

¹ Im Lebenszyklusmodell von [Kuh07] entspricht dieser Schritt der *Tailoringstufe 1*.

3.2 Reichhaltige Vorgehensmodelle

Insbesondere ist eine so erstellte Variante nicht konform zum V-Modell XT im Sinne eines Standards.

Änderung: Die vorhandenen Inhalte des V-Modell XT werden direkt bearbeitet und verändert. Soll die Konformität zum Standard gewahrt bleiben, so müssen eventuelle Änderungen am Standard manuell mit den eigenen Änderungen in Einklang gebracht werden.

Erweiterungsmodell: die vorhandenen Inhalte des V-Modell XT bleiben unangetastet. Der Prozessingenieur legt ein Erweiterungsmodell an und definiert in diesem über Änderungsoperationen die Abweichungen vom Standard.

Der Neuaufbau der Vorgehensmodellinhalte verwendet nur das V-Modell XT Metamodell. Inhaltlich hat das Ergebnis nichts mehr mit dem Standard zu tun. Ebenso verhält es sich mit der Änderung der Vorgehensmodellinhalte: hierbei handelt es sich um einen Neuaufbau eines Vorgehensmodells unter Rückgriff auf vorhandene Inhalte.

Soll die Konformität zum Standard gewahrt bleiben (ohne eventuelle Änderungen im Standard manuell in der eigenen Variante nachpflegen zu müssen), müssen die Standardinhalte unverändert bleiben und Anpassungen in Form von Änderungsoperationen getrennt vom Standard V-Modell XT definiert werden. Derartige Modifikationen können in einem sogenannten *Erweiterungsmodell* definiert werden. Da das Erweiterungsmodell selbst auf dem Metamodell des V-Modell XT basiert, kann der Anpasser in diesem vollständige Module inhaltlich ausgestalten und diese über die *Änderungsoperationen* mit den Inhalten des Basismodells verknüpfen. Folgende Änderungsoperationen werden im Erweiterungsmodell unterstützt:

- Umbenennung von Modellelementen
- Ergänzung von Beschreibungstexten
- Ersetzung von Beschreibungstexten
- Verschiebung von Modellelementen
- Änderung von Beziehungen

Abbildung 3.2 veranschaulicht den Zusammenhang zwischen Vorgehensmodell und Erweiterungsmodell: das Vorgehensmodell definiert Inhalte – im Beispiel der Abbildung 3.2 das Artefakt *Projekthandbuch* und die Rolle *Projektleiter*. Im Erweiterungsmodell drückt eine Änderungsoperation *RolleUmbenennen* aus, dass die Rolle *Projektleiter* in der organisationsspezifischen Ausprägung *Projektoverantwortlicher* heißen soll.

Ebenfalls wesentlicher Bestandteil der organisationsspezifischen Anpassung ist die Definition der *Tailoringstruktur*. Durch diese wird festgelegt, welche Optionen im Tailoring zur Verfügung stehen und welche Auswirkungen auf Inhalte des Modells diese haben.

Variabilitätspunkte

Die beim Tailoring des V-Modell XT involvierten Modellelemente und ihr Zusammenhang sind in Abbildung 3.3 dargestellt.

Die Struktur und Mächtigkeit des Tailorings ist durch das Metamodell des V-Modell XT vorgegeben. Zur Beschreibung der Kriterien des Tailorings dienen die Elemente *Projekttyp*, *Projekttypvariante* und *Projektmerkmal*. Durch einen Projekttyp wird eine grobe Klassifizierung des Projekts vorgenommen. Projekttypen schließen sich gegenseitig aus, d.h. von den möglichen Projekttypen ist beim

3 Anpassungsfähigkeit von Vorgehensmodellen

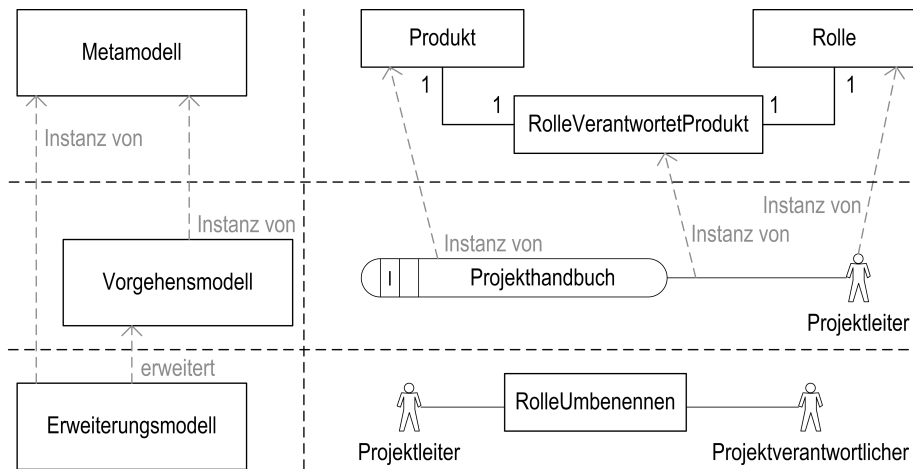


Abbildung 3.2: Vorgehensmodell und Erweiterungsmodell im V-Modell XT (entnommen aus [KTF09])

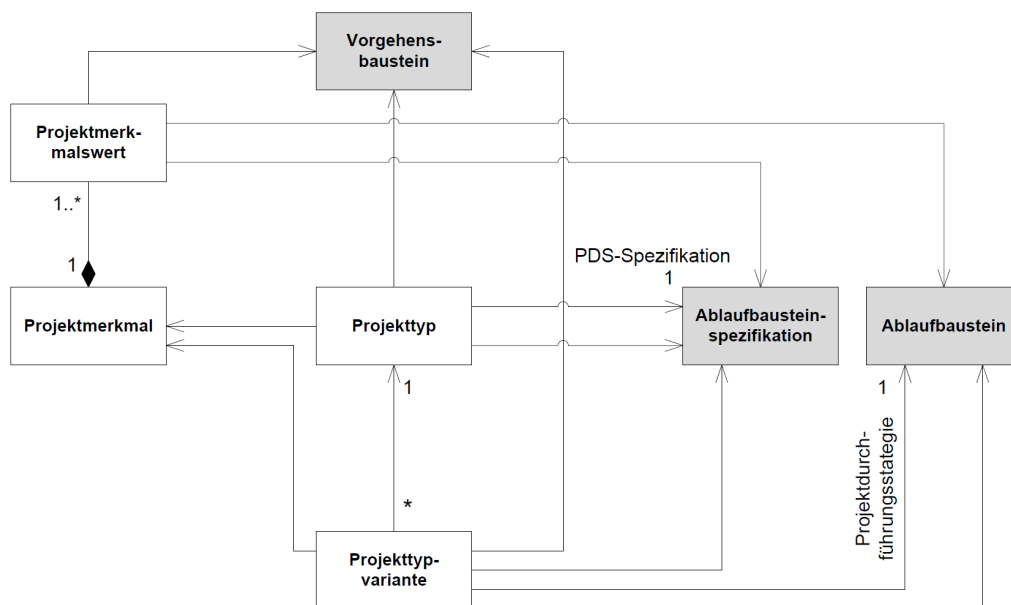


Abbildung 3.3: Metamodell zum Tailoring im V-Modell XT [TK09]

3.2 Reichhaltige Vorgehensmodelle

Tailoring genau einer zu wählen. Abhängig von der Wahl des Projekttyps lässt sich das Projekt durch die Wahl einer Projekttypvariante enger klassifizieren. Auch Projekttypvarianten schließen sich gegenseitig aus. Sowohl Projekttypen als auch Projekttypvarianten können Projektmerkmale bedingen - d.h. sind Projekttyp und Projekttypvariante gewählt, lässt sich durch die Wertbelegung von Projektmerkmalen das vorliegende Projekt weiter charakterisieren.

Projekttypen, Projekttypvarianten, und die Werte der Projektmerkmale entscheiden über den Ein- oder Ausschluss von *Vorgehensbausteinen* und *Ablaufbausteinen* in der projektspezifischen Ausprägung des V-Modell XT.

Ein Vorgehensbaustein fasst inhaltlich zusammengehörige Produkte, Themen der Produkte, Aktivitäten und weitere zu den jeweiligen Produkten gehörige Modellelemente zusammen. Zudem können die Inhalte eines Vorgehensbausteins Inhalte aus anderen Vorgehensbausteinen erweitern. So können beispielsweise in einem Vorgehensbaustein inhaltlich zusammengehörige Themen zusammengefasst sein, die bei Wahl des entsprechenden Tailoring-Kriteriums in Produkte aus anderen Vorgehensbausteinen einfließen.

Ein Ablaufbaustein definiert eine zeitliche Abfolge von Meilensteinen. Durch die Wahl der Ablaufbausteine können beim Tailoring alternative Strategien zur Projektdurchführung gewählt werden.

Tailoring auf Ebene atomarer Vorgehensmodellinhalte ist nicht möglich. Sollte etwa ein einzelnes Produkt durch eine Tailoring-Entscheidung in der projektspezifischen Ausprägung ein- oder ausgeschlossen werden, so müsste ein Vorgehensbaustein definiert werden, welcher nur das eine Produkt beinhaltet. Wie in Abschnitt 7.3 ausgeführt, ist der gemeinsame Ein- oder Ausschluss der Inhalte eines Vorgehensbausteins sinnvoll, da andernfalls Konventionen des V-Modell XT verletzt werden könnten (beispielsweise „jedes Produkt hat eine bearbeitende Aktivität“).

Tailoring: Das Tailoring wird vom Projektleiter zu Beginn des Projekts durchgeführt. Durch die Wahl eines *Projekttyps*, einer *Projekttypvariante* und die Wertbelegung der *Projektmerkmale* wird eine grobe Anpassung des Vorgehensmodells an die Situation des Projekts möglich. Abbildung 3.4 zeigt den V-Modell XT Projektassistenten mit zwei Projekttypen und zwei Projekttypvarianten. Im unteren Bereich der Abbildung (unter „Übersicht“) sind die von der Wahl betroffenen Vorgehensbausteine dargestellt.

Da Projektmerkmale nicht belegt werden müssen um ein exportfähiges Tailoring durchzuführen, bestehen die Arbeitsschritte zum Tailoring oft nur in der Wahl des Projekttyps und der -Variante (siehe auch Abbildung 3.5).

Diskussion

Das „XT“ in V-Modell XT steht für Extreme Tailoring. Die Fähigkeit zur flexiblen Anpassung war bei der Entwicklung des V-Modell XT eines der wesentlichen Ziele.

Hinsichtlich der Fähigkeit zur organisationsspezifischen Ausgestaltung und -Anpassung sucht das V-Modell XT seinesgleichen: die Auslagerung organisationsspezifischer Inhalte in ein Erweiterungsmodell, die Änderungsoperationen und der V-Modell XT Editor machen das V-Modell XT gut geeignet zur Entwicklung einer organisationsspezifischen Variante.

Das eigentliche Tailoring ist dagegen relativ beschränkt und statisch. Der Prozessingenieur ist bei der Definition von Kriterien zur Projektcharakterisierung

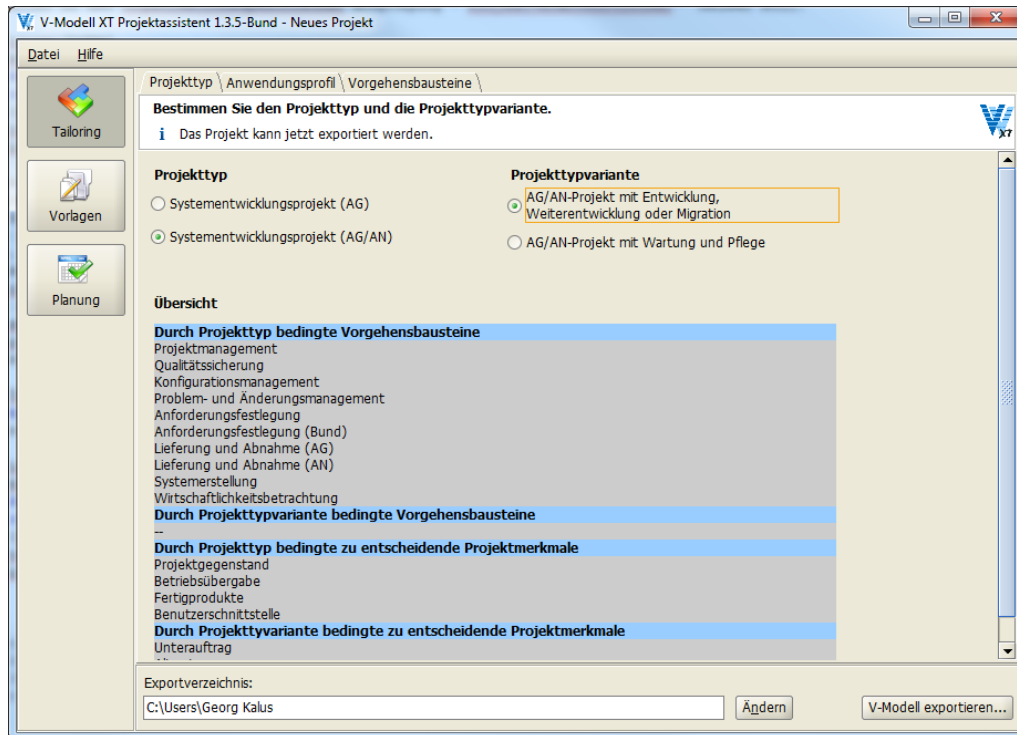


Abbildung 3.4: V-Modell XT Projektassistent

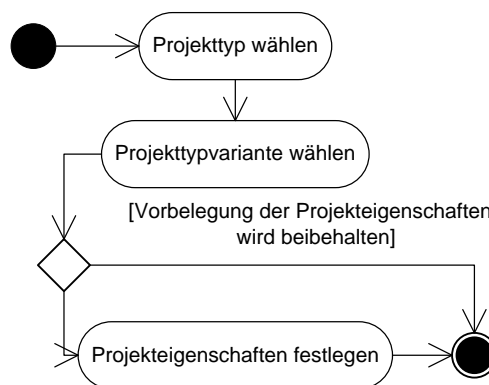


Abbildung 3.5: Schritte des Tailorings im V-Modell XT

3.2 Reichhaltige Vorgehensmodelle

an die durch das Metamodell vorgegebene Struktur, bestehend aus Projekttyp, Projekttypvariante und Projektmerkmal, gebunden. Auch die Art der Entscheidungen ist vorgegeben: von Projekttypen und Projekttypvarianten ist jeweils genau einer (bzw. eine) zu wählen. Bei den Projektmerkmalen kann genau ein Wert aus n Werten gewählt werden.

So wie das Tailoring im V-Modell XT realisiert ist, erlaubt es zu Projektbeginn eine grobe Anpassung an projektspezifische Umstände. Durch die feste Struktur ist es nicht skalierbar: da beim Tailoring nur zwei Entscheidungen (für Projekttyp und Projekttypvariante) getroffen werden können plus der Belegung einer Reihe von Projektmerkmalen ist eine feingranulare Auswahl von Vorgehensmodellinhalten nicht möglich – auch wenn sich die Variabilitätspunkte prinzipiell dafür eignen würden (vergleiche hierzu insbesondere auch die Fallstudie zum feature-basierten Tailoring des V-Modell XT in Abschnitt 7.3).

Das Tailoring im V-Modell XT beinhaltet keine Möglichkeit, Abhängigkeiten zwischen den Tailoring-Kriterien zu modellieren. Gegenseitiger Ausschluss von Projektmerkmalen oder automatische Wahl eines Merkmals bei Wahl eines anderen lassen sich nicht darstellen.

Neben dieser Beschränkung der Ausdrucksmächtigkeit ist der Prozessingenieur bei der Definition des Tailorings auf sich gestellt und auf seine umfassende Kenntnis des gesamten Vorgehensmodells angewiesen: Konflikte in der Modellierung treten erst beim Export eines projektspezifischen V-Modells auf.

3.2.2 Das Eclipse Process Framework

Das Eclipse Process Framework (EPF) [Ecl11a] ist eine Implementierung des Software Process Engineering Metamodel (SPEM) [OMG08]. SPEM ist ein umfangreiches Metamodell zur Beschreibung von Vorgehensmodellen. Die Inhalte sind hier in sogenannten *Methodenbibliotheken* zusammengefasst, welche zu einem kompletten Vorgehensmodell kombiniert werden können.

Anpassungsfähigkeit

Die Grundphilosophie der Anpassung besteht bei EPF nicht im Maßschneidern eines Standardvorgehens, sondern in der Kombination und Konfiguration von Bausteinen. Eine Trennung wie beim V-Modell XT zwischen der organisationspezifischen Anpassung, bei welcher das Vorgehensmodell bearbeitet und verändert wird, und dem Tailoring, bei dem das Vorgehensmodell projektspezifisch konfiguriert wird, gibt es nicht. Generell ist bei Erstellung eines Vorgehens basierend auf EPF ein tiefes Verständnis des Prozesses, der zur Verfügung stehenden Bausteine und der zu verwendenden Werkzeuge nötig.

Abbildung 3.6 stellt die wesentlichen Elemente der Vorgehensmodellbeschreibung in EPF. Ein „Methodenpaket“ besteht im Wesentlichen aus den statischen Inhalten *Work Products*, *Roles* und *Tasks* und den Inhalten, welche Abläufe definieren, namentlich *Capability Pattern*, *Delivery Process* und *Activity Guidance*, also Hinweise und „Anleitungen“ können sowohl statische als auch Ablaufinhalte beschreiben und erweitern.

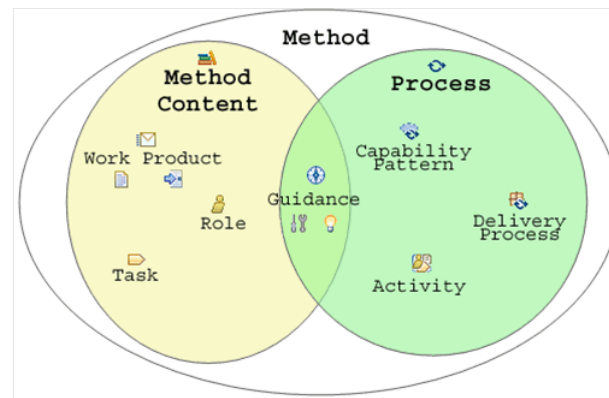


Abbildung 3.6: Kernelemente von EPF

Laut [Bal07] lassen sich bei der Anpassung bzw. der Erstellung eines mit Hilfe von EPF beschriebenen Vorgehensmodells folgende typische Szenarien unterscheiden:

Es sind keine eigenen Inhalte erforderlich. In diesem Fall kann auf einen Standardprozess zurückgegriffen werden. Ein Beispiel für ein fertiges, in EPF beschriebenes Vorgehensmodell ist der Open Unified Process (OpenUP) [Ecl11b].

Standardinhalte werden in einer eigenen Konfiguration benötigt. Lassen sich die Anforderungen an das Vorgehensmodell über Standardinhalte des Eclipse Process Frameworks abbilden, so genügt es, eine eigene Konfiguration zu erstellen, welche die erforderlichen Inhalte geeignet kombiniert.

Standardinhalte sollen um Hinweise erweitert werden. Wenn sich die Anforderungen an das Vorgehensmodell prinzipiell über Standardinhalte darstellen lassen und lediglich um organisations- oder projektspezifische Hinweise erweitert werden müssen, beispielsweise zur Verwendung einer Methode oder Beschreibungstechnik, so können derartige Hinweise in Form von *Guidance* in einem EPF Plugin verwaltet werden. Die Inhalte des Plugins können sich dabei auf existierende Inhalte beziehen. Damit ist die Erweiterung von Standardinhalten möglich, ohne diese selbst verändern zu müssen.

Ein eigener Projektlebenszyklus wird benötigt. Wenn die Abläufe in der Organisation bzw. im Projekt sich nicht durch einen bereits von EPF definierten Prozess abbilden lassen, kann ein eigener Prozess in Form geeigneter *Capability Patterns*, *Activities* und eines *Delivery Process* erstellt werden. Ähnlich wie eigene Hinweise, z. B. zur Verwendung einer Methode, wird der Prozess in einen EPF Plugin definiert und bezieht sich auf die vorhandenen, statischen Inhalte.

Gemeinsam haben alle Szenarien (bis auf das erste), dass die Durchführung der Anpassung im EPF Composer (eine typische Arbeitsfläche dieses Werkzeugs ist in Abbildung 3.7 dargestellt) durchgeführt wird. Dieses Werkzeug ist auf der Eclipse Plattform realisiert. Eine Trennung der Werkzeuge für die organisationspezifische Anpassung und Entwicklung eines Vorgehensmodells und dem Tailoring zu Projektbeginn wie beim V-Modell XT gibt es beim Eclipse Process Framework nicht, da auch die Anpassungsschritte nicht wie beim V-Modell XT als Aktivitäten aufgefasst werden, welche im Regelfall von unterschiedlichen Rollen durchgeführt werden. Das *Tailoring* zu Projektbeginn entspricht in EPF

3.2 Reichhaltige Vorgehensmodelle

am ehesten dem zweiten Szenario – der Konfiguration bestehender Inhalte. Allerdings erfolgt die Erstellung einer Konfiguration nicht anhand einer (durch das Vorgehensmodell vorgegebenen) Menge von Kriterien, wie im V-Modell XT beispielsweise dem Projekttyp und den Projektmerkmalen.

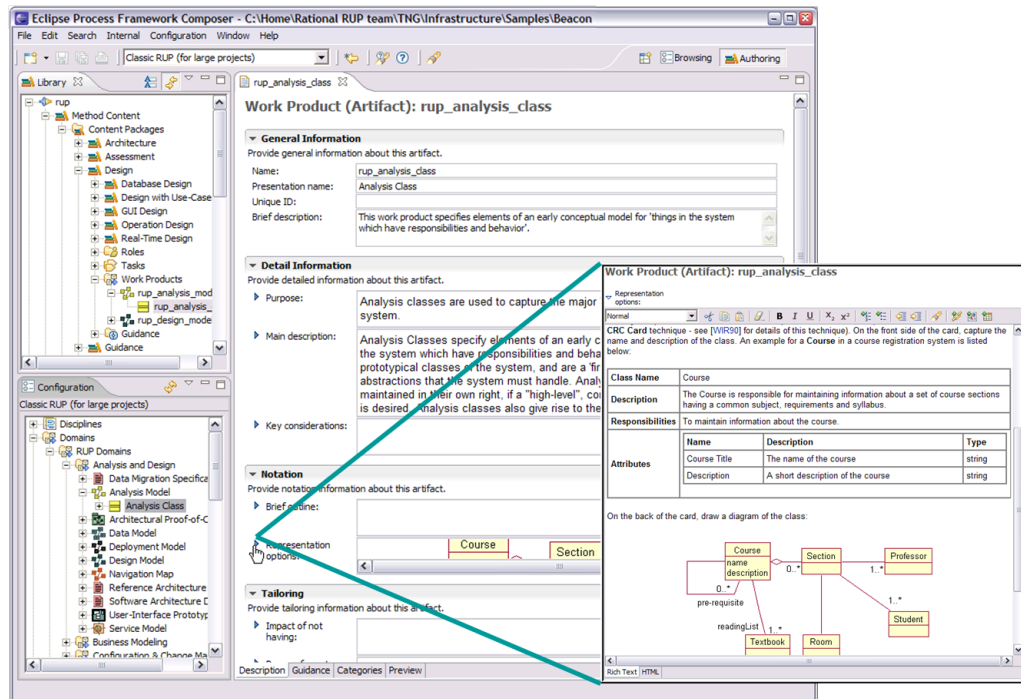


Abbildung 3.7: EPF Composer

Auch das zugrunde liegende Metamodell SPEM [OMG08] macht keine detaillierten Aussagen zur projektspezifischen Anpassung. Zwar wird der Bedarf anerkannt, wie folgendes Zitat verdeutlicht:

For example, a software development project that develops an application from scratch performs development tasks such as „Find Actors and Use Cases,“ „Develop Vision,“ or „Design Use Case“ similarly to a project that extends an existing software system. However, the two projects will perform the Tasks at different points in time with a different emphasis, i.e., they will perform the steps of these tasks differently, assume different inputs, and perhaps apply individual variations and additions.

[OMG08]

Wie Projekte, die Anwendungen von Grund auf neu erstellen und solche, die ein bestehendes Softwaresystem erweitern, unterschieden werden und welche Schlüsse aus dieser Unterscheidung für das Vorgehensmodell zu ziehen sind, macht die SPEM Spezifikation nicht. Das Beispiel wird in der SPEM Spezifikation lediglich dazu verwendet, die lose Kopplung zwischen Vorgehensmodellinhalten zu motivieren ohne darauf einzugehen, wie mit diesem Bedarf konkret umzugehen ist.

Variabilitätspunkte

Der zentrale Variabilitätspunkt im Eclipse Process Framework ist der Methodenbaustein. Methodenbausteine ähneln konzeptionell den Vorgehensbausteinen aus dem V-Modell XT: sie fassen zusammengehörige Modellinhalte, wie Arbeitsergebnisse, Rollen und Aktivitäten, zusammen. Wie bei den Vorgehensbausteinen im V-Modell XT können sich die Inhalte eines Methodenbausteins auf Inhalte anderer Methodenbausteine beziehen und diese erweitern, modifizieren oder ersetzen.

Ein Vorgehensmodell entsteht durch die Kombination von Methodenbausteinen in einer *Konfiguration*. Eine schematische Darstellung der Konfiguration von Methodenbausteinen findet sich in Abbildung 3.8. Die rot eingefärbten Methodenbausteine repräsentieren in der Abbildung die durch die Konfiguration gewählte Untermenge der gesamten Methodenbibliothek.

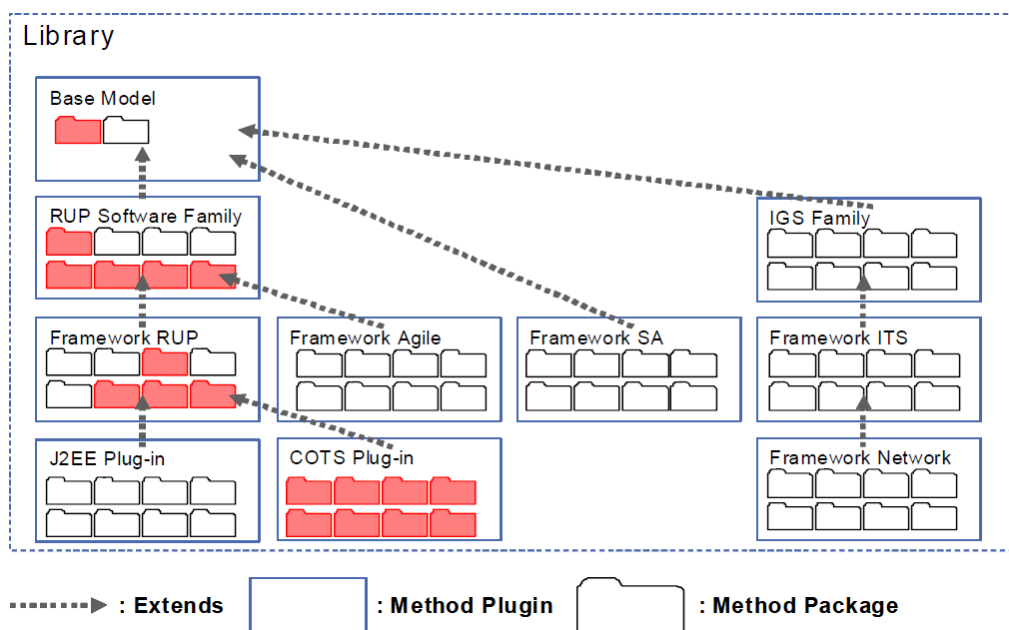


Abbildung 3.8: Konfiguration von Methodenbausteinen in SPEM [OMG08]

Da mit Hilfe einer Konfiguration Inhalte für eine projektspezifische Instanz eines Vorgehensmodells flexibel ein- und ausgeschlossen werden können, ist die Variabilität eines auf EPF (oder SPEM) basierenden Vorgehensmodells sehr hoch. Da EPF (und SPEM) aber keine Modellstrukturen zur Charakterisierung eines Projekts (wie die Projekttypen, Projekttypvarianten und Projektmerkmale im V-Modell XT) beinhalten, ist bei der Erstellung einer Konfiguration eine gute Kenntnis der Inhalte der Methodenbausteine nötig.

Diskussion

Die Möglichkeiten zur Anpassung eines EPF-basierten Vorgehensmodells decken die ganze Bandbreite ab von der Erstellung eines komplett neuen Vorgehens bis hin zur Verwendung von fertigen Modellen, wie dem OpenUP. Allerdings erfordert auch schon ein vergleichsweise kleiner „Eingriff“, wie beispielsweise die Erstellung einer projektspezifischen Konfiguration, ein hohes

3.2 Reichhaltige Vorgehensmodelle

Maß an Expertenwissen, sowohl was das Werkzeug angeht (da der EPF Composer auf der Eclipse Plattform basiert, erinnert dieses Werkzeug stark an ein „Entwicklerwerkzeug“), als auch in Bezug auf das Vorgehensmodell selbst. Wie die Vorgehensmodell-Bausteine in geeigneter Weise kombiniert werden können bzw. müssen, damit ein vollständiges, gültiges und sinnvolles Vorgehensmodell entsteht, erfordert Kenntnis des Metamodells und des Modells.

Ein Tailoring basierend auf einer festgelegten Reihe von Kriterien, welche das zu beginnende Projekt charakterisieren, ist in EPF nicht vorgesehen. Bis auf die im folgenden Abschnitt 3.2.3 beschriebene schweizerische Projektführungsmethode ist kein auf dem Eclipse Process Framework basierendes Vorgehensmodell bekannt, welches konkrete Kriterien für die Anpassung und Konsequenzen der Kriterien für die Inhalte des Vorgehensmodells vereinbart. Betrachtet man beispielsweise die Dokumentation des Open Unified Process [Ecl11b], werden dort auch (nur) die vier oben genannten Anpassungsszenarien beschrieben. Die Auswahl geeigneter Bausteine zur Zusammenstellung einer projektspezifischen Variante des OpenUP obliegt damit vollständig dem Prozessingenieur (es kann angenommen werden, dass aufgrund der Komplexität dieser Tätigkeit und des erforderlichen Expertenwissens eine derartige Anpassung nicht vom Vorgehensmodellanwender – beispielsweise dem Projektleiter – zu Projektbeginn für jedes Projekt individuell durchgeführt wird).

Im Vergleich zu den in Abschnitt 3.2.1 diskutierten Anpassungsschritten des V-Modell XT entspricht die Anpassung eines auf EPF basierenden Vorgehensmodells der Erstellung einer organisationspezifischen Variante und nicht dem in Definition 3 vereinbarten Verständnis von Tailoring eines Vorgehensmodells.

3.2.3 HERMES - Die schweizerische Projektführungsmethode

Das Vorgehensmodell HERMES [Inf04] zur Durchführung von Projekten in der Informations- und Kommunikationstechnik wurde von der schweizerischen Bundesverwaltung entwickelt und ist – analog zum V-Modell XT in Deutschland – das Vorgehensmodell der Schweiz für entsprechende Projekte. HERMES steht für „Handbuch der elektronischen Rechenzentren des Bundes, Methode für die Entwicklung von Systemen“.

HERMES ist ein an das Wasserfallmodell angelehntes, ergebnisorientiertes Vorgehensmodell. Die Inhalte von HERMES sind nach SPEM strukturiert. Damit entspricht die grundsätzliche Struktur der Inhalte von HERMES der aus Abbildung 3.6.

Anpassungsfähigkeit

Das Werkzeug zur Bearbeitung und Anpassung von HERMES ist der sogenannte „HERMES PowerUser“, ein Werkzeug, welches auf dem EPF Composer aufsetzt. Einem Anpasser von HERMES stehen damit alle Möglichkeiten von EPF zur Verfügung (vergleiche Abschnitt 3.2.2). Besonderes Merkmal von HERMES PowerUser ist, dass es um eine Benutzeroberfläche zum Tailoring erweitert wurde. Diese ist in Form eines Assistenten realisiert, welcher den Benutzer durch die Schritte des Tailorings führt (zwei Masken des Tailoring-Assistenten von HERMES sind in Abbildung 3.9 dargestellt). Folgende Schritte sind beim Tailoring von HERMES durchzuführen:

Wahl von Projekttyp und -kategorie: Bei den Projekttypen wird zwischen den Typen Systementwicklung und Systemadaption unterschieden. Für jeden

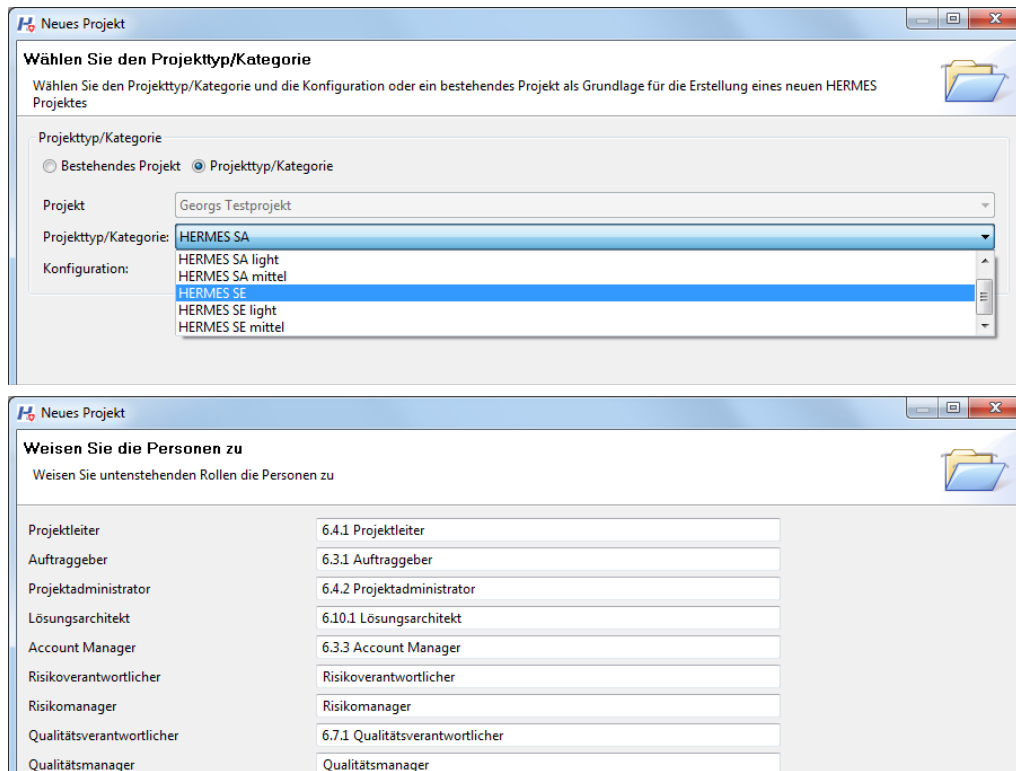


Abbildung 3.9: Tailoring in HERMES PowerUser

der beiden Projekttypen gibt es drei Ausprägungen von HERMES, deren Wahl sich nach der Projektkategorie richtet (zur Projektkategorie siehe unten). Zusätzlich zu diesen Angaben kann durch die Wahl einer Konfiguration das äußere Erscheinungsbild der Ergebnisvorlagen (z. B. eine eigene Corporate Identity) gesteuert werden.

Angabe von Vorlageneigenschaften: Bei der Erstellung eines neuen Projekts in HERMES PowerUser werden für die Ergebnisse Word-Vorlagen generiert. Diese lassen sich durch Angabe von Vorlageneigenschaften parametrisieren. Durch die Vorlageneigenschaften kann beispielsweise definiert werden, welcher Organisations- und Projektname in den Vorlagen erscheinen soll oder der Name der Person, welche die Rolle „Projektleiter“ einnimmt, kann angegeben werden.

Feintailoring auf Basis des Arbeitsstrukturplans: Im Feintailoring lassen sich anhand der Phasen, Aktivitäten und Arbeitsschritte diejenigen Inhalte des Vorgehensmodells auswählen, die initial erstellt werden sollen. Mit anderen Worten: hier lässt sich festlegen, welche Vorlagen nach Abschluss des Assistenten generiert werden. Einen Einfluss auf die Inhalte der Vorgehensmodellokumentation hat das Feintailoring nicht.

Für die Wahl einer Projektkategorie (jeweils *vollständig*, *mittel*, und *klein* für beide Projekttypen) bietet HERMES umfassende Hinweise. Die Einstufung eines Projekts erfolgt auf Basis der Beurteilung eines der Faktoren Wichtigkeit, Größe oder Risiko (siehe Tabelle 3.1).

Für die Einschätzung der Größe werden die in Tabelle 3.2 dargestellten Werte vorgeschlagen.

3.2 Reichhaltige Vorgehensmodelle

Kategorie	Wichtigkeit	Größe	Risiko
vollständig	hoch	groß	hoch
mittel	mittel	mittel	mittel
klein	niedrig	klein	niedrig

Tabelle 3.1: Projektkategorien von HERMES

Größe	Aufwand	Teamgröße	Investition
groß	> 1000 PT	> 5 Mitarbeiter	> 2,0 Mio. CHF
mittel	≤ 1000 PT	≤ 5 Mitarbeiter	≤ 2,0 Mio. CHF
klein	≤ 100 PT	≤ 2 Mitarbeiter	≤ 0,2 Mio. CHF

Tabelle 3.2: Projektgröße nach HERMES

Variabilitätspunkte

Da HERMES auf EPF (und damit auf SPEM) basiert, sind die Variabilitätspunkte bei diesem Vorgehensmodell die gleichen wie in Abschnitt 3.2.2 beschrieben. Wesentlicher Unterschied zum reinen EPF bei HERMES ist die Wahl vordefinierter Konfigurationen basierend auf der Abfrage der in den Tabellen 3.1 und 3.2 beschriebenen Kriterien zur Charakterisierung des vorliegenden Projekts.

Diskussion

Weder das V-Modell XT noch EPF geben dem Projektleiter Hinweise, wie das Vorgehensmodell an die Bedürfnisse des zu beginnenden Projekts angepasst werden kann. HERMES hebt sich hier durch die in den Tabellen 3.1 und 3.2 dargestellten Anhaltspunkte zur Einstufung des Projekts ab. Allerdings beziehen sich die Hinweise lediglich auf die Wahl der Projektkategorie. Bei zwei Projekttypen und jeweils drei Kategorien ergeben sich lediglich sechs verschiedene Tailoringergebnisse. In dem in HERMES als Feintailoring bezeichneten Schritt lässt sich lediglich steuern, welche Vorlagen zu Beginn des Projekts angelegt werden sollen. Die Ausprägung des Vorgehensmodells ändert sich dadurch nicht.

Gegenüber dem EPF Composer in seiner unmodifizierten Form bietet das Werkzeug HERMES PowerUser mit dem Tailoring-Assistenten einige Erleichterungen für den Vorgehensmodell-Anwender, welcher sich nicht mit den technischen Feinheiten des Vorgehensmodells auseinandersetzen kann und will. Da es sich bei HERMES PowerUser um eine Erweiterung des EPF Composer handelt und HERMES selbst auf EPF basiert, bieten sich zur Erstellung einer organisationsspezifischen Variante alle Möglichkeiten, die auch mit anderen auf EPF basierenden Modellen vorhanden sind (siehe Abschnitt 3.2.2).

3.2.4 SE Book

Das SE Book ist das hausinterne Vorgehensmodell zur Softwareentwicklung der T-Systems International GmbH. Das Metamodell des SE Book liegt in seiner Philosophie zwischen dem des V-Modell XT und SPEM. Wie im V-Modell XT sind die Inhalte im Modell streng hierarchisch gruppiert. Wie in SPEM definiert das SE Book Ein- und Ausgabeartefakte für Aktivitäten.

Anpassungsfähigkeit

Analog zum V-Modell XT Projektassistenten hat das SE Book mit dem *Process Suite Generator* ein Werkzeug zur Erstellung einer projektspezifischen Ausprägung des Vorgehensmodells. Es erlaubt die Generierung der projektspezifischen Dokumentation des Vorgehensmodells in Form einer Webseite oder als PDF Dokument und erstellt Vorlagen für die Arbeitsergebnisse. Im April 2010 beinhaltet der *Process Suite Generator* vier Schalter mit denen das Vorgehensmodell an die Projektsituation angepasst werden konnte.

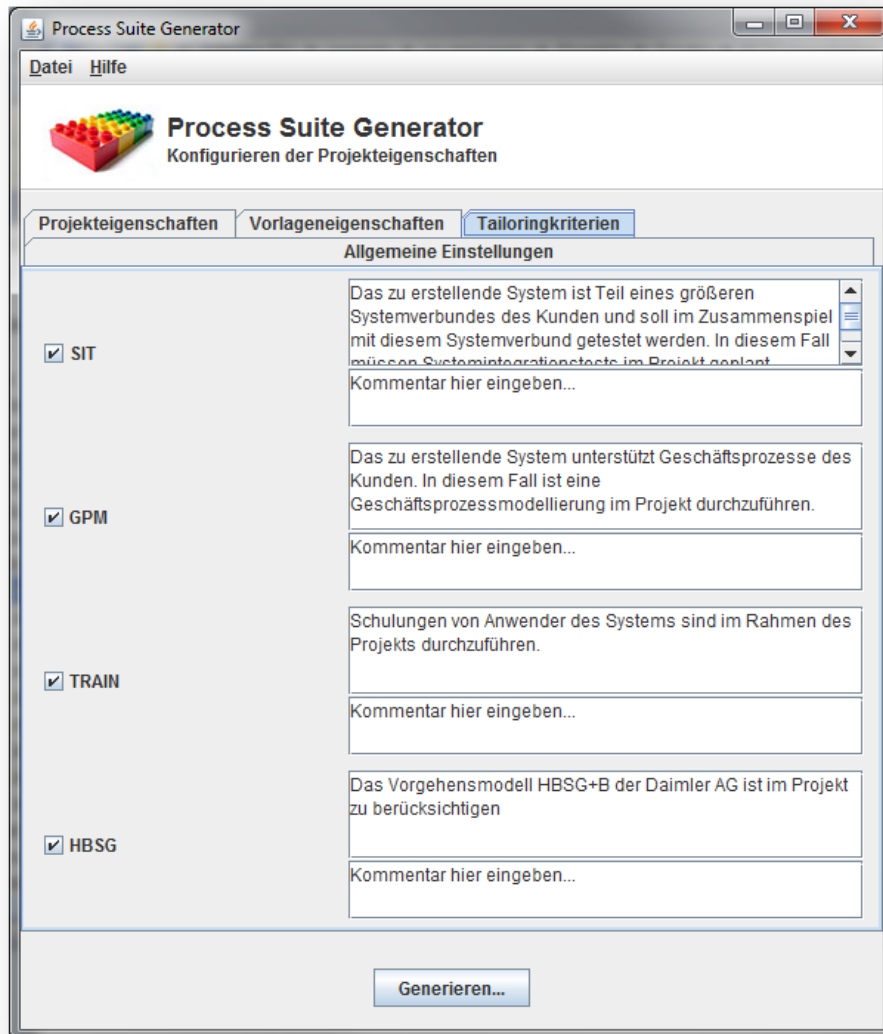


Abbildung 3.10: Tailoring im SE Book Process Suite Generator

Wie in Abbildung 3.10 dargestellt, sind die vier Schalter im Einzelnen:

SIT Im Rahmen des Projekts sind Systemintegrationstests durchzuführen. Ist diese Option gewählt, werden die für Systemintegrationstests relevanten Aktivitäten und Artefakte in das Ergebnis-Vorgehensmodell mit aufgenommen.

GPM Im Rahmen des Projekts ist Geschäftsprozessmodellierung durchzuführen.

3.2 Reichhaltige Vorgehensmodelle

ren. Auch hier werden bei Wahl der Option einige Aktivitäten und Artefakte in die projektspezifische Ausprägung des Vorgehensmodells aufgenommen.

TRAIN Sind im Projekt Anwenderschulungen durchzuführen, werden bei Wahl dieser Option entsprechende Aktivitäten und Artefakte in das Ergebnis-Vorgehensmodell eingeschlossen.

HBSG Die T-Systems International GmbH führt eine Reihe von Projekten für den Kunden Daimler AG durch. Dieser Kunde stellt spezielle Anforderungen an die Form der zu erstellenden Arbeitsergebnisse. Handelt es sich bei dem zu startenden Projekt um eines für die Daimler AG, können über die Option HBSG spezielle, den Kundenanforderungen genügende, Vorlagen in das Ergebnis-Vorgehensmodell aufgenommen werden.

Die Wahl eines der Schalter führt zum Einschluss der mit dem entsprechenden Kriterium assoziierten Inhalte in die projektspezifische Ausprägung. Zwischen den Kriterien können keine Querbezüge, wie Abhängigkeit oder gegenseitiger Ausschluss, hergestellt werden.

Neben den vier Schaltern zum Tailoring im Process Suite Generator ist zu Beginn eines Entwicklungsprojekts nach SE Book die sogenannte *Tailoring-Checkliste* auszufüllen. Hierbei handelt es sich um eine Excel-Tabelle, welche alle vom SE Book vorgesehenen Arbeitsergebnisse beinhaltet. Zu jedem Arbeitsergebnis ist in der Tailoring-Checkliste anzugeben, ob dieses *Vorlagenkonform* (die vom SE Book bereitgestellte Vorlage wird verwendet), *Strukturkonform* (die vom SE Book vorgesehene Kapitelstruktur wird eingehalten) oder *Inhaltskonform* (die vom SE Book vorgesehenen Inhalte finden sich wieder) erstellt wird.

Variabilitätspunkte

Das Modellelement zur Repräsentation der Schalter beim Tailoring des SE Book ist das `Label`. Fast alle inhaltstragenden Modellelemente im SE Book können Verweise auf ein Label tragen. Damit können prinzipiell fast alle Modellelemente ein Variabilitätspunkt im SE Book sein. Die Semantik eines Label ist folgende: hat ein Modellelement keinen Verweis auf ein Label, so ist es Teil jeder möglichen Ausprägung des SE Book. Verweist das Modellelement auf ein Label, so ist es nur dann Teil der projektspezifisch angepassten Instanz, wenn das Label beim Tailoring im Process Suite Generator ausgewählt wird. Hat ein Modellelement Verweise auf mehrere Label, ist es Teil der projektspezifischen Instanz des SE Book, sobald mindestens eins der referenzierten Label beim Tailoring gewählt ist.

Abbildung 3.11 zeigt einen Ausschnitt aus dem SE Book Metamodell: das SE Book ist in sogenannte *Bücher* unterteilt. Jedes der Bücher besteht aus einer Reihe von Modulen, die ihrerseits Artefakte und Aktivitäten (im Metamodell *UseCase* genannt) beinhalten. Aktivitäten verweisen wiederum auf eine Reihe von benötigten Eingabeartefakten und eine Reihe von produzierten Ausgabeartefakten. Bis auf das übergeordnete Buch können alle im Metamodell-Ausschnitt in Abbildung 3.11 dargestellten Modellelemente einen Verweis auf Label tragen und dadurch Gegenstand des Tailorings werden.

Diskussion

Das SE Book bietet einige grundlegende Mechanismen zur Anpassung des Vorgehensmodells an die Projektsituation. Durch Verwendung des Process Suite

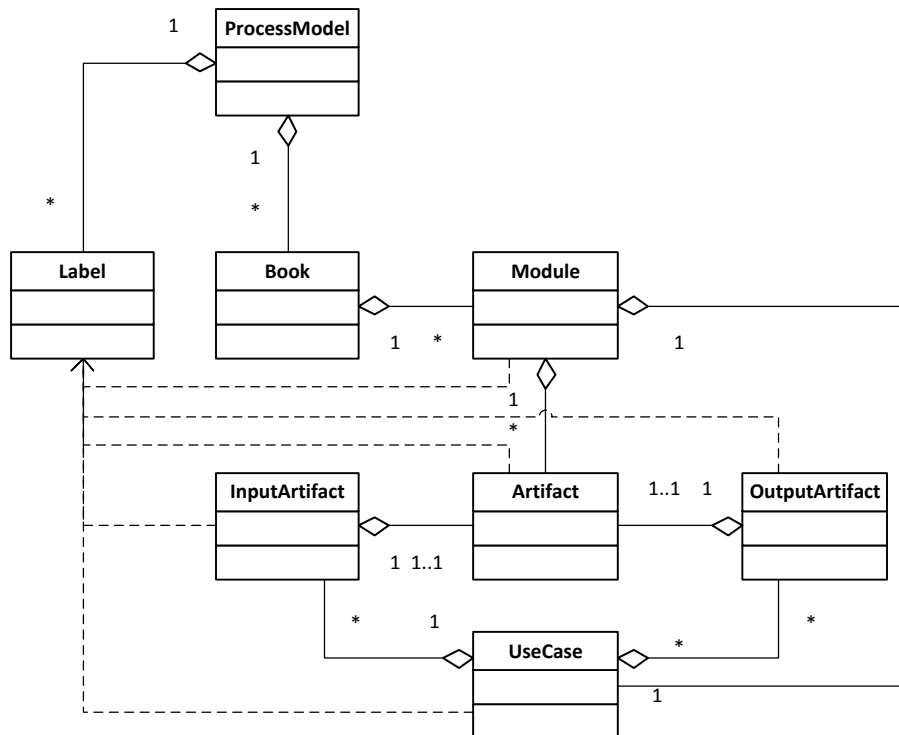


Abbildung 3.11: Zusammenhang zwischen Labels und Modellelementen im SE Book

Generator bei der Erstellung einer projektspezifischen Instanz des Vorgehensmodells ist die Anpassung für den Anwender des SE Book leicht durchzuführen. Für ein feingranulares Tailoring auf Basis einzelner Arbeitsergebnisse muss auf eine Excel-Tabelle ausgewichen werden. Da die Excel-Tabelle keinen Bezug zum zugrunde liegenden Modell hat, kann in diesem Schritt die Einhaltung der durch das SE Book vorgegebenen Regeln zum Tailoring nicht sichergestellt werden.

In der in Abschnitt 7.2 beschriebenen Fallstudie wenden wir den in der vorliegenden Arbeit entwickelten Ansatz zum feature-basierten Tailoring auf das SE Book an.

3.3 Leichtgewichtige Vorgehensweisen

Im folgenden Abschnitt befassen wir uns mit der Anpassungsfähigkeit sogenannter leichtgewichtiger Vorgehensweisen. Das Gewicht einer Vorgehensweise bemisst sich hierbei an dem in diesem Kapitel eingangs beschriebenen Produkt aus Anzahl der Elemente (wie Arbeitsergebnisse, Rollen und Aktivitäten) und dem Maß für das durch die Vorgehensweise vorgeschriebene „Zeremoniell“ (vergleiche Abschnitt 3.1).

Als leichtgewichtig im Sinne dieses Gewichtsmaßes werden insbesondere Extreme Programming [Bec03, BA04], die Crystal Methodologies [Coc04, Coc06], Scrum [Sch04] und Feature Driven Development [PF02] bezeichnet.

Gemein ist diesen Vorgehensweisen, dass sie darauf abzielen, ein Projekt durch möglichst kurze Entwicklungszyklen tolerant und robust gegenüber Veränderungen (beispielsweise volatile Anforderungen) zu machen. Vom Ablauf eines

3.3 Leichtgewichtige Vorgehensweisen

Projekts sind diese Vorgehensweisen deshalb im Bereich der iterativen oder evolutionären Vorgehensmodelle anzusiedeln.

Weiterhin zeichnen sich die leichtgewichtigen Vorgehensweisen dadurch aus, dass sie die am Projekt beteiligten Menschen und ihre Zusammenarbeit ins Zentrum der Betrachtung stellen, denn: „The quality of the people on a project, and their organization and management, are more important factors in success than are the tools they use or the technical approaches they take.“ [Bro95].

Die Philosophie der leichtgewichtigen und „veränderungstoleranten“ Vorgehensweisen schlägt sich nieder im 2001 formulierten *Agile Manifesto*:

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Agile Alliance im Agile Manifesto [BBvB⁺01]

Die im Agile Manifesto festgehaltenen Werte führen dazu, dass leichtgewichtige (synonym: agile) Methoden eine minimale Menge von Arbeitsergebnissen, Aktivitäten und Rollen definieren und zudem meist eine Reihe von Praktiken, insbesondere zur Zusammenarbeit im Team, beinhalten. Um einen Eindruck für den Umfang und die Schwerpunktsetzung der leichtgewichtigen Methoden zu bekommen, sind in Tabelle 3.3 die wesentlichen Elemente von Extreme Programming (XP), Crystal Clear (CC), Scrum und Feature Driven Development (FDD) zusammengefasst.

Ohne auf die Inhalte und deren Bedeutung näher einzugehen, ist anhand der Aufstellung in Tabelle 3.3 leicht zu erkennen, dass die etwa von Scrum vorgegebenen Inhalte kaum sinnvoll im Sinne eines Tailorings² anzupassen sind. Scrum beschreibt zudem nur Rollen, Aktivitäten und Ergebnisse, die unmittelbar mit dem Vorgehen selbst in Bezug stehen. Es werden keine Vorgaben für Artefakte zur Angebotserstellung, Anforderungserhebung, Spezifikationen, Tests, usw. gemacht. Im Feature Driven Development sind einige der Aktivitäten, beispielsweise *Study Documents*, als optional markiert. Dies ist so zu verstehen, dass Dokumente bei Vorhandensein studiert werden sollten. Auch wenn die Aktivität als optional markiert ist, wäre es sicher nicht im Sinne der Vorgaben des Feature Driven Development, diese Aktivität bei der projektspezifischen Anpassung aus der Dokumentation des Vorgehensmodells auszuschließen.

Generell ergibt sich aus der Aufstellung in Tabelle 3.3 folgendes Bild: Inhalte der leichtgewichtigen Vorgehensbeschreibungen, die potenziell einer projektspezifischen Anpassung unterworfen sein könnten, wie Artefakte und Aktivitäten, sind so minimal gehalten, dass hier Mechanismen der projektspezifischen Anpassung – beispielsweise zur Reduktion des bürokratischen Aufwands oder der Dokumentation der Vorgehensweise – nicht sinnvoll angewendet werden können. In [Coc04] wird als ein Ziel bei der Entwicklung der Methodensammlung Crystal Clear angegeben, eine Vorgehensweise für kleine Softwareentwicklungsprojekte zu definieren, die „gerade so ausreichend“ ist („bare-

² entsprechend Abschnitt 2.2.2

Meth.	Zentrale Inhalte
XP	<p>5 Werte: Kommunikation, Einfachheit, Rückmeldung, Mut, Respekt</p> <p>14 Prinzipien: Menschlichkeit, Wirtschaftlichkeit, beidseitiger Vorteil, Selbstgleichheit, Verbesserungen, Vielfältigkeit, Reflexion, Lauf, Gelegenheiten wahrnehmen, Redundanzen vermeiden, Fehlschläge hinnehmen, Qualität, kleine Schritte, akzeptierte Verantwortung</p> <p>13 Hauptpraktiken: Räumlich zusammen sitzen, informativer Arbeitsplatz, Team, Pair-Programming, energiegeladene Arbeit, entspannte Arbeit, Storys, wöchentlicher Zyklus, quartalsweiser Zyklus, 10-Minuten-Build, kontinuierliche Integration, Test-First-Programmierung, inkrementelles Design</p> <p>11 Begleitpraktiken: richtige Kundeneinbeziehung, inkrementelles Deployment, Team-Konstanz, schrumpfende Teams, ursächliche Analysen, gemeinsamer Code, Codierung und Testen, eine zentrale Codebasis, tägliches Deployment, verhandelbarer, vertraglicher Funktionsumfang, Zahlen-pro-Nutzung</p>
CC	<p>8 Rollen: Sponsor, Team, Coordinator, Business Expert, Lead Designer, Designer-Programmer, Tester, Writer</p> <p>22 Artefakte: Mission Statement, Team Structure and Conventions, Reflection Workshop Results, Project Map, Release Plan, Project Status, Risk List, Iteration Plan, Viewing Schedule, Actor-Goal List, Use Cases and Requirements, User Role Model, Architecture Description, Screen Drafts, Common Domain Model, Design Sketches, Source Code, Migration Code, Tests, Packaged System, Bug Reports, User Help Text</p> <p>5 Strategien: Exploratory 360°, Early Victory, Walking Skeleton, Incremental Rearchitecture, Information Radiators</p> <p>9 Techniken: Methodology Shaping, Reflection Workshop, Blitz Planning, Delphi Estimation Using Expertise Rankings, Daily Stand-up Meetings, Essential Interaction Design, Process Miniature, Side-by-Side Programming, Burn Charts</p>
Scrum	<p>5 Rollen: Product Owner, Entwicklungsteam, ScrumMaster, Customer, Management</p> <p>3 Zeremonien: Sprint Planning, Sprint Review, Daily Scrum</p> <p>3 Artefakte: Product Backlog, Sprint Backlog, Burndown Chart</p>
FDD	<p>5 Prozesse: Develop an Overall Model, Build a Features List, Plan By Feature, Design By Feature, Build By Feature</p> <p>10 Rollen: Project Manager, Modelling Team, Chief Architect, Chief Programmer, Business, Development Manager, Feature List Team, Planning Team, Domain Expert, Feature Team</p> <p>24 Aktivitäten: Form the Modelling-Team, Domain Walk-through, Study Documents, Develop the Model, Refine the Overall Object Model, Write Model Notes, Internal and External Assessment, Form the Features List Team, Build Features List, Form the Planning Team, Determine the Development Sequence, Assign Business Activities to Chief Programmers, Assign Classes to Developers, Self Assessment, Form the Feature Team, Develop the Sequence Diagrams, Refine the Object Model, Write Class and Method Prologues, Design Inspection, Implement Classes and Methods, Code Inspection, Unit Test, Promote to the Build, Code Inspection and Unit Test</p>

Tabelle 3.3: Überblick über Inhalte leichtgewichtiger Methoden

3.4 Zusammenfassung

ly sufficient“). Eine ähnliche Aussage findet sich zum Feature Driven Development: Dieses sei „just enough process to ensure scalability and repeatability and encourage creativity and innovation all along the way“ [Hig02].

Ein Großteil der Inhalte leichtgewichtiger Vorgehensweisen besteht aus Prinzipien, Praktiken, Strategien, usw., die dann ihre Wirkung entfalten können, wenn sie in der Kultur der Organisation oder des Teams verankert sind und *gelebt* werden. Für diese Inhalte spielt es keine Rolle, ob sie in der Dokumentation der Vorgehensweise enthalten sind oder nicht.

Die „Erstellung“ einer projektspezifischen Instanz einer leichtgewichtigen Vorgehensweise besteht also im Wesentlichen in der Verankerung der Vorgaben in der Kultur und Arbeitsweise der Organisation bzw. des Teams und der Art und Weise, *wie* die Praktiken der Vorgehensweise gelebt werden: „Agile methods derive much of their agility by relying on the tacit knowledge embodied in the team, rather than writing the knowledge down in plans“ [Boe02].

Auch für agile Vorgehensweisen wird die Notwendigkeit der projekt- und team-spezifischen Ausgestaltung anerkannt und zahlreiche Arbeiten beschäftigen sich mit geeigneten Verfahren [BA04, MKRS⁺08, Coc00b, Coc06]. In der Tat scheint es z.B. bei Extreme Programming die Regel zu sein, dass in der Praxis nicht alle der Praktiken angewendet werden [ASRW02, Gre01, Sch01]. Verfahren zur projektspezifischen Anpassung agiler Vorgehensweisen, wie z. B. [MKRS⁺08], beschreiben im Kern eine weitere Praktik, welche das Team bei der Auswahl geeigneter Werte, Regeln und Praktiken aus der eigentlichen Sammlung unterstützen soll.

Ein Charakteristikum aller agilen Vorgehensweisen ist die Verankerung einer Retrospektive, beispielsweise nach jeder Iteration [ASRW02]. Die Retrospektive betrifft insbesondere auch die Vorgehensweise im Projekt. Diese soll abhängig von den Gegebenheiten im Projekt kontinuierlich kritisch überprüft und gegebenenfalls angepasst werden.

Fazit: Genau wie reichhaltige Vorgehensmodelle müssen auch agile Vorgehensweisen für die konkrete Projektsituation und das Team adaptiert werden. Dieser Vorgang lässt sich – wie in der Regel die komplette Methodenbeschreibung – kaum sinnvoll formalisieren, ohne die Philosophie der Vorgehensbeschreibung ad absurdum zu führen. Der in unserer Arbeit entwickelte Ansatz des feature-basierten Tailorings birgt dann einen Mehrwert, wenn das Vorgehensmodell „genügend“ Inhalte hat, die eine projektspezifische Anpassung, insbesondere der Dokumentation und der das Vorgehensmodell begleitenden Materialien (Vorlagen für Ergebnisse, usw.), rechtfertigen. Da agile Vorgehensweisen bewusst minimal gehalten sind, wird er sich auf diese Klasse von Vorgehensweisen nicht sinnvoll anwenden lassen.

3.4 Zusammenfassung

In diesem Kapitel haben wir die Anpassungsfähigkeit ausgewählter Vorgehensmodelle beleuchtet. Neben einer Betrachtung der Mächtigkeit der projektspezifischen Anpassung haben wir das dahinter stehende Konzept und Verfahren analysiert. Als Vertreter reichhaltiger Vorgehensmodelle haben wir die Fähigkeit zur projektspezifischen Anpassung des V-Modell XT, von Vorgehensmodellen basierend auf dem Eclipse Process Framework und des SE Book der T-Systems International GmbH eingehend analysiert. Als Spezialfall der auf dem Eclipse

Process Framework basierenden Vorgehensmodelle haben wir die schweizerische Projektführungsmethode HERMES betrachtet. Zu jedem Vorgehensmodell haben wir insbesondere untersucht, welche Möglichkeiten der Beschreibung der Projektcharakteristika das Verfahren zur Anpassung erlaubt und welche Variabilitätspunkte das Anpassungsverfahren vorsieht.

Wie in Abschnitt 2.5 ausgeführt, lassen sich die Verfahren zur Erstellung einer projektspezifischen Vorgehensmodellinstanz in solche einteilen, die für ein konkretes Projekt eine neue Methode *konstruieren* und solche, die eine gegebene Obermenge an Vorgaben für die jeweilige Projektsituation *konfigurieren*. Der in der vorliegenden Arbeit entwickelte Ansatz des feature-basierten Tailorings fällt in die zweite Kategorie.

Im zweiten Teil dieses Kapitels haben wir die Anpassungsfähigkeit leichtgewichtiger Methoden untersucht. Wie haben hierzu die Inhalte populärer agiler Vorgehensweisen analysiert und festgestellt, dass diese sich nicht für eine projektspezifische Anpassung im Sinne der vorliegenden Arbeit eignen.

Struktur von Vorgehensmodellen

In Kapitel 3 haben wir die Anpassungsfähigkeit von Vorgehensmodellen untersucht. Besonderes Augenmerk lag bei dieser Untersuchung auf der Sicht des Vorgehensmodellanwenders, wie der Anpassungsvorgang durchzuführen ist und welche Möglichkeiten zur Anpassung sich dem Anwender zur Erstellung einer projektspezifischen Vorgehensmodellausprägung bieten. In diesem Kapitel analysieren wir die innere Struktur von Vorgehensmodellen und leiten daraus eine Formalisierung für Vorgehensmodelle ab. Die Formalisierung dient als Grundlage für das feature-basierte Tailoring.

Am Ende dieses Kapitels wird der Leser ein Verständnis für die innere Struktur von Vorgehensmodellen haben und mit der in dieser Arbeit verwendeten Formalisierung von Vorgehensmodellen vertraut sein. Damit ist das Fundament für die Entwicklung des feature-basierten Tailorings gelegt.

Übersicht

4.1	Überblick	88
4.2	Analyse: Das V-Modell XT Metamodell	89
4.3	Analyse: das Metamodell SPEM	97
4.4	Analyse: Formalisierung leichtgewichtiger Methoden .	103
4.5	Ein einfaches Meta-Metamodell für Vorgehensmodelle	105
4.6	Tailoring neu formuliert	112
4.7	Zusammenfassung	113

4.1 Überblick

Wie in Abschnitt 2.2.1 ausgeführt, beschreibt ein Vorgehensmodell Artefakte, Rollen, Aktivitäten, Abläufe, Werkzeuge, Methoden und Hilfsmittel zur Durchführung eines (Software-)Entwicklungsprojekts. Diese Inhalte stehen über Beziehungen miteinander in Verbindung. Eine Festlegung im Vorgehensmodell könnte etwa lauten: „der Projektleiter ist verantwortlich für die Erstellung des Projektplans“ oder „für den Meilenstein *System spezifiziert* ist das Artefakt *Systemspezifikation* vorzulegen“.

Die Beziehungen zwischen Vorgehensmodell-Inhalten haben üblicherweise eine Richtung: eine Beziehung könnte etwa lauten: „die Rolle Projektleiter verantwortet die Erstellung des Produkts Projektplan“. Zudem können die Beziehungen unterschiedliche Ausprägungen haben. Im V-Modell XT wird bei Beziehungen zwischen Rolle und Produkt beispielsweise unterschieden zwischen „Rolle verantwortet Produkt“ und „Rolle wirkt mit an Produkt“. Auch in SPEM-basierten Vorgehensmodellen werden Beziehungen über den Typ der Beziehung qualifiziert (vgl. auch Abschnitt 4.3.3).

These: Die Struktur eines Vorgehensmodells lässt sich als gerichteter Graph ohne Mehrfachkanten darstellen, wobei die Knoten die Inhaltsträger sind und die Kanten die Beziehungen zwischen Inhalten herstellen.

Abbildung 4.1 veranschaulicht den Aufbau dieses Kapitels.

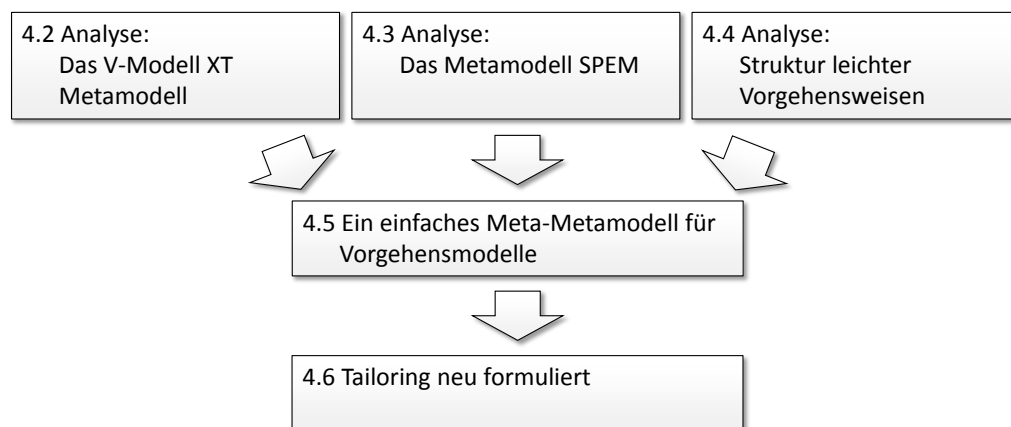


Abbildung 4.1: Struktur des Kapitels 4

In den folgenden Abschnitten 4.2 und 4.3 analysieren wir exemplarisch das Metamodell des V-Modell XT und das SPEM Metamodell, um die einleitende These zu überprüfen. Im Abschnitt 4.4 untersuchen wir die Struktur leichtgewichtiger Methodensammlungen, denen kein formal definiertes Metamodell unterliegt. Mit dem Begriff „leichtgewichtig“ beziehen wir uns auf das in Abschnitt 3.1 eingeführte Gewichtsmaß für Vorgehensmodelle. Wir zeigen, dass sich derartige Methodensammlungen zwar prinzipiell ebenfalls als gerichteter Graph darstellen lassen, sich aber aufgrund der Philosophie dieser Vorgehensweisen kaum geeignete Variabilitätspunkte für die projektspezifische Anpassung identifizieren lassen.

Vor dem Hintergrund der Analyse der inneren Struktur von Vorgehensmodellen nehmen wir in Abschnitt 4.5 eine Formalisierung des Vorgehensmodellbegriffs vor und formulieren die Aufgabe des Tailorings in Abschnitt 4.6 neu: die

projektspezifische Anpassung eines Vorgehensmodells zu Projektbeginn kann basierend auf der Formalisierung als Konfiguration der Graphstruktur des Vorgehensmodells aufgefasst werden.

4.2 Analyse: Das V-Modell XT Metamodell

Das V-Modell XT liegt auf technischer Ebene in Form einer XML-Datei vor. Die Struktur der XML-Datei wird bestimmt durch ein V-Modell XT XSD-Schema¹, welches – soweit mit den Mitteln von XSD möglich – das V-Modell XT Metamodell repräsentiert. Nicht alle Eigenschaften und Konventionen des Metamodells lassen sich mit Hilfe von XSD ausdrücken. Beispielsweise ist die Konvention, dass zu jedem Produkt im V-Modell XT eine bearbeitende Aktivität definiert wird, nicht mit den Mitteln von XSD auszudrücken. Für die physikalische Ablage des V-Modell XT in einer XML-Datei muss dieses als Baum dargestellt werden. Im linken Bereich der Abbildung 4.2 ist die Baumstruktur des V-Modell XT im V-Modell XT Editor zu erkennen. Der Baum im Editor entspricht dabei exakt der Baumstruktur der zugrunde liegenden XML-Datei.

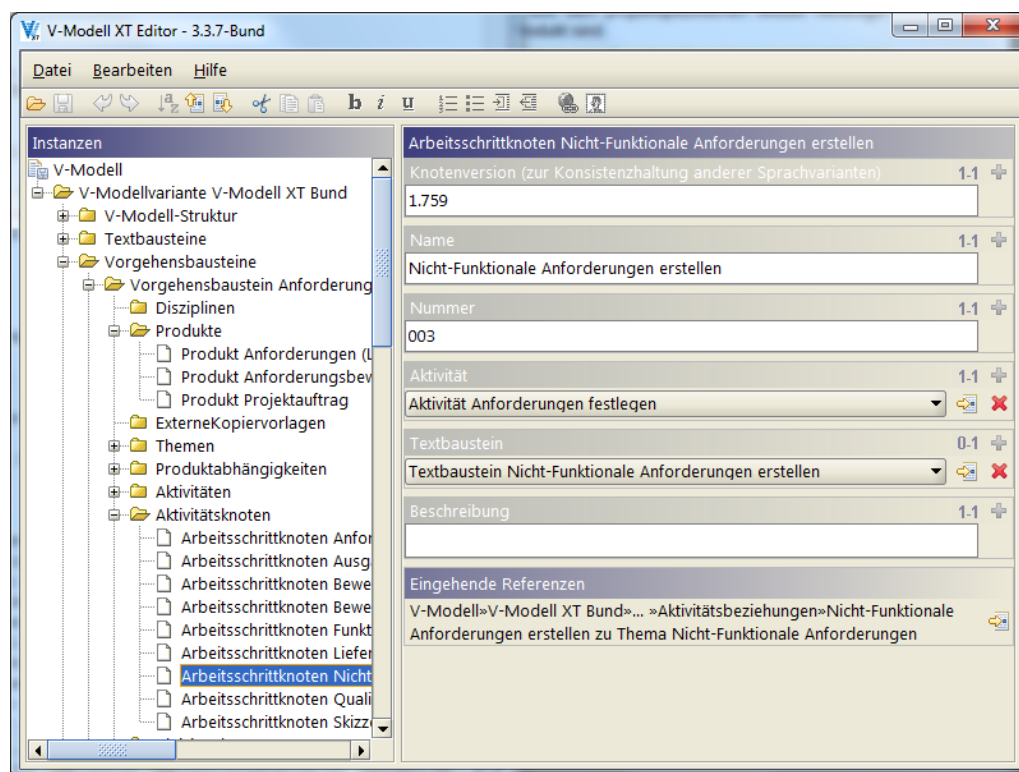


Abbildung 4.2: V-Modell XT Inhalte als Baum im V-Modell XT Editor

4.2.1 Elemente und Beziehungen

Wie sich aus Abbildung 4.2 erahnen lässt, dient die Baumstruktur im Wesentlichen einer hierarchischen Ordnung der Inhalte des Vorgehensmodells. Die eigentliche *Struktur* des Vorgehensmodells ergibt sich aus Beziehungen zwischen den Knoten im Baum über Hierarchiegrenzen hinweg.

¹ XSD steht für XML Schema Definition [BM04].

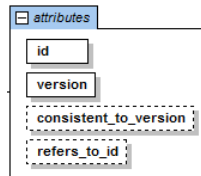


Abbildung 4.3: Attribute referenzierbarer Elemente im V-Modell XT

Referenzierbare Elemente

Zur Darstellung der Beziehungen werden im V-Modell XT Metamodell alle potenziell an einer Beziehung teilnehmenden Modellelemente mit einem modellweit eindeutigen Identifikator ausgezeichnet. In Abbildung 4.3 findet sich der entsprechende Ausschnitt aus dem XML-Schema des V-Modell XT: jedes potenziell an einer Beziehung teilnehmende (referenzierbare) Modellelement hat vier Attribute:

id Dieses Attribut beinhaltet den modellweit eindeutigen Identifikator des Elements. Über diesen Identifikator kann das Element referenziert werden.

version Dieses Attribut beinhaltet eine Versionsnummer des Modellknotens und wird bei der (Weiter-)Entwicklung des V-Modell XT benutzt – insbesondere zur Sicherstellung der Konsistenz bei verschiedenen Sprachversionen des V-Modell XT.

consistent_to_version Dieses (optionale) Attribut dient der Unterstützung verschiedener Sprachen im V-Modell XT.

refers_to_id Dieses (optionale) Attribut dient ebenfalls der Internationalisierung des Modells.

In der XML-Datei des V-Modells sieht ein referenzierbares Element mit diesen vier Attributen so aus, wie in Abbildung 4.4 für das Modellelement „Nicht-funktionale Anforderungen“ vom Typ `Thema` exemplarisch dargestellt. Wie in der Abbildung zu erkennen, wurden die beiden optionalen Attribute `consistent_to_version` und `refers_to_id` für dieses Modellelement nicht angegeben.

```
<Thema id="1383e1079f7ec427" version="1.760">
  <!-- Name Start 1383e1079f7ec427 -->
  <Name>Nicht-Funktionale Anforderungen</Name><!-- 1383e1079f7ec427 -->
```

Abbildung 4.4: Beispielausprägung der Attribute referenzierbarer Elemente

Beziehungen

Beziehungen zwischen (referenzierbaren) Modellelementen kommen im V-Modell XT in zwei Ausprägungen vor: als Attribut eines Modellelements (in XML dargestellt als nicht referenzierbares Kindelement) oder in Form von Beziehungsknoten, welche ihrerseits referenzierbare Modellelemente sind und oftmals zusätzliche Inhalte, wie z. B. eine textuelle Beschreibung der Beziehung, tragen (können).

Beziehungen als Attribut

Abbildung 4.5 stellt abstrakt die Beziehungen zwischen Modellelementen dar, bei denen die Beziehung als Teil eines der an der Beziehung teilnehmenden Elemente realisiert ist.

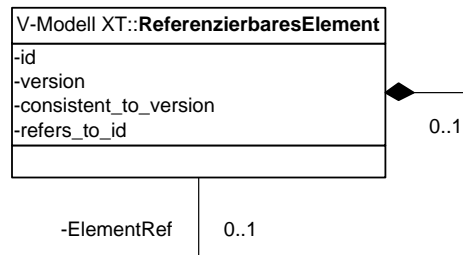


Abbildung 4.5: Beziehung als Attribut

Ein konkretes Beispiel für eine als Attribut definierte Beziehung ist die zwischen Produkt und der Disziplin des Produkts (Disziplinen im V-Modell XT stellen eine thematische Gruppierung der Produkte dar). Abbildung 4.6 stellt einen entsprechenden Ausschnitt aus der XML Datei des V-Modell XT für das Produkt Ausschreibungskonzept und die zugehörige Disziplin Ausschreibungswesen (Vergabeakte) dar. Wie zu erkennen, entspricht der Wert des Attributs link des Kindknotens DisziplinRef vom Produkt Ausschreibungskonzept dem Wert des Attributs id der Disziplin mit dem Namen Ausschreibungswesen (Vergabeakte). Da der Knoten DisziplinRef keine id hat, ist er selbst nicht referenzierbar.

```
<Produkt Initial="Nein" Extern="Nein" Produktvorlage="Ja" id="1200812492085e5f" version="1.1">
  <!-- Name Start 1200812492085e5f -->
  <Name>Vergabevermerk</Name><!-- 1200812492085e5f -->
  <!-- Name Ende 1200812492085e5f -->
  <!-- Nummer Start 1200812492085e5f -->
  <Nummer>001</Nummer><!-- 1200812492085e5f -->
  <!-- Nummer Ende 1200812492085e5f -->
  <!-- DisziplinRef Start 1200812492085e5f -->
  <DisziplinRef link="31e712492069ec3" /><!-- 1200812492085e5f -->
  <!-- DisziplinRef Ende 1200812492085e5f -->

  <Disziplinen id="e80f12492054462" version="1.1">
  <!-- Disziplin Start e80f12492054462 -->
  <Disziplin id="31e712492069ec3" version="1.1">
  <!-- Name Start 31e712492069ec3 -->
  <Name>Ausschreibungswesen (Vergabeakte)</Name><!-- 31e712492069ec3 -->
  <!-- Name Ende 31e712492069ec3 -->
```

Abbildung 4.6: Beispielausprägung einer Beziehung als Attribut

Beziehungselemente

Der überwiegende Teil der Beziehungen im V-Modell XT ist über explizite Beziehungselemente dargestellt. Vor allem in Hinblick auf Anpassung bieten Beziehungselemente Vorteile gegenüber als Attribut modellierten Beziehungen, da sie erstens referenzierbar sind und da sie zweitens die Inhalte des Vorgehensmodells von der durch die Beziehungen aufgespannten Graphstruktur entkoppeln. Die Struktur wird dadurch selbst variabler Anteil des Vorgehensmodells.

Abbildung 4.7 zeigt wieder einen Ausschnitt aus der XML Datei des V-Modell XT: dargestellt ist eine Beziehung vom Typ RolleWirktMitBeiProdukt. Neben

4.2 Analyse: Das V-Modell XT Metamodell

einem Namen hat dieser Beziehungstyp ein optionales Beschreibungsfeld (erkennbar an den Kommentaren im XML). Die wesentlichen Elemente sind die fünf auf `-Ref` endenden Elemente. Im dargestellten Beispiel setzen sie die Rolle *Projektleiter* mit fünf Produkten in Beziehung, an denen diese Rolle mitwirkt.

```
<RolleWirktMitBeiProdukt id="39ee124a47fd3d7" version="1.1">
<!-- Name Start 39ee124a47fd3d7 -->
<Name>Projektleiter</Name><!-- 39ee124a47fd3d7 -->
<!-- Name Ende 39ee124a47fd3d7 -->
<!-- Beschreibung Start 39ee124a47fd3d7 -->
<!-- Beschreibung Ende 39ee124a47fd3d7 -->
<!-- IstUnverzichtbar Start 39ee124a47fd3d7 -->
<!-- IstUnverzichtbar Ende 39ee124a47fd3d7 -->
<!-- RolleRef Start 39ee124a47fd3d7 -->
<RolleRef link="4871f6d2b11e07" /><!-- 39ee124a47fd3d7 -->
<!-- RolleRef Ende 39ee124a47fd3d7 -->
<!-- ProduktRef Start 39ee124a47fd3d7 -->
<ProduktRef link="15d1b1249209be2f" /><!-- 39ee124a47fd3d7 -->
<ProduktRef link="423f1249207f3e1" /><!-- 39ee124a47fd3d7 -->
<ProduktRef link="7ec412492093383" /><!-- 39ee124a47fd3d7 -->
<ProduktRef link="13819124920b00a5" /><!-- 39ee124a47fd3d7 -->
<ProduktRef link="161ee124920ca389" /><!-- 39ee124a47fd3d7 -->
<!-- ProduktRef Ende 39ee124a47fd3d7 -->
</RolleWirktMitBeiProdukt><!-- 39ee124a47fd3d7 -->
```

Abbildung 4.7: Beispielausprägung einer Beziehung als Element

Das dem Beispiel aus Abbildung 4.7 zugrunde liegende Metamodell ist in Abbildung 4.8 dargestellt. Da das Element `RolleWirktMitBeiProdukt` selbst ein referenzierbares Element ist, kann es bei der Anpassung des Modells direkt manipuliert werden. Weiterhin erlauben explizite Beziehungselemente im Gegensatz zu Beziehungen, welche als Attribut modelliert sind, die Darstellung von $1 : n$ und $m : n$ Beziehungen.

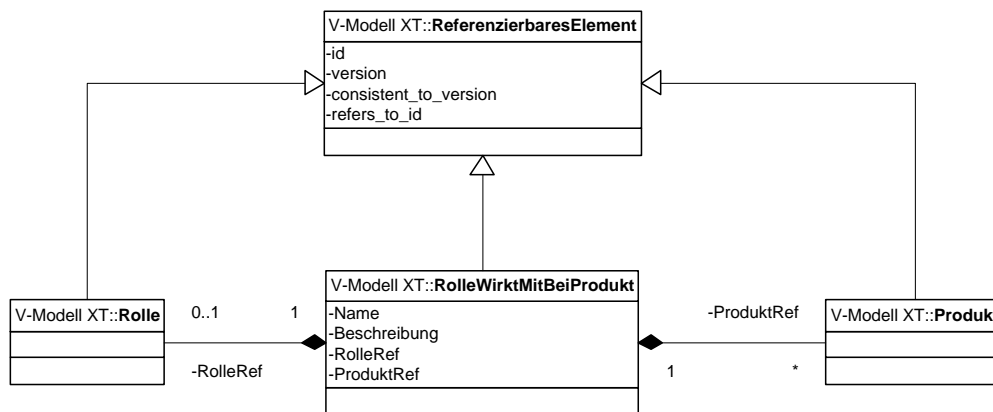


Abbildung 4.8: Beziehung als Modellelement

4.2.2 Paketstruktur

Das Metamodell des V-Modell XT lässt sich in eine logische Paketstruktur zergliedern [TK09].

Das Paket *Statik*² im V-Modell XT umfasst im Wesentlichen die Teilmodelle

² Der Vollständigkeit halber seien auch die anderen Pakete erwähnt:

- das Paket *Basis* beschreibt die Basisstruktur des Vorgehensmodells;
- das Paket *Dynamik* beinhaltet die Modellelemente zur Steuerung des Projektablaufs und

Produktmodell, Aktivitätsmodell und das *Rollenmodell*. Produkte und Aktivitäten, die inhaltlich zusammengehören, werden dabei zu *Vorgehensbausteinen* zusammengefasst. In Bezug auf die statischen Anteile des Vorgehensmodells dienen Vorgehensbausteine zusätzlich als diejenigen Einheiten, die durch das Tailoring im Ergebnismodell ein- oder ausgeschlossen werden können. Dadurch wird sichergestellt, dass zusammengehörige Produkte und Aktivitäten nur zusammen aus dem Ergebnismodell ausgeschlossen (bzw. eingeschlossen) werden können. Vorgehensbausteine sind also der zentrale Variabilitätspunkt im V-Modell XT. Neben den Vorgehensbausteinen gibt es im Paket *Dynamik* noch die Variabilitätspunkte *Ablaufbaustein* und *Ablaufbausteinspezifikation*. Diese dienen der projektspezifischen Auswahl einer geeigneten Projektdurchführungsstrategie, also einer zeitlichen Anordnung der Meilensteine im Projekt und damit indirekt der Ergebniserstellung.

Rollen liegen inhaltlich quer zu den Ergebnisstrukturen. In der Regel wird eine Rolle aus mehreren Vorgehensbausteinen referenziert. Diese sind deshalb im V-Modell XT nicht in die Vorgehensbausteine gekapselt. Rollen werden allerdings in den Vorgehensbausteinen über Beziehungselemente eingebunden.

Abbildung 4.9 zeigt die Bestandteile eines Vorgehensbausteins. Die wesentlichen Inhalte befinden sich dabei unter den Knoten *Disziplinen, Produkte, Themen* und *Aktivitäten*. Beziehungen zwischen den Inhalten sind unterhalb des Knotens *Beziehungen* gesammelt.

Die Beziehungen zwischen Vorgehensbaustein und den Knoten *Disziplinen, Produkte, Themen*, usw. sind alle der Art *ist Teil von*. Betrachten wir beispielsweise den Knoten *Produkte* näher, so stellen wir fest, dass sich ein Produkt in weitere Bestandteile aufgliedert (vergleiche hierzu Abbildung 4.10). Die Elemente *Name, Nummer, Sinn_und_Zweck* bilden die Blätter des Strukturbaums. Von besonderem Interesse sind die auf *-Ref* endenden Elemente. Diese stellen eine Beziehung zu anderen Modellinhalten her. Beispielsweise kann über das Element *TextbausteinRef* eine Beziehung zu einem Modellelement vom Typ *Textbaustein* hergestellt werden. Das Element *TextbausteinRef* spannt eine gerichtete Kante vom Quell-Produkt zum Ziel-Textbaustein auf.

Wie bereits dargelegt, werden die meisten Beziehungen im V-Modell XT über *Beziehungselemente* dargestellt. Durch die „Auslagerung“ von Beziehungen in vollwertige Modellelemente (also Knoten im Graphen) können Beziehungen durch Typisierung des Beziehungselements qualifiziert werden. Zudem werden Beziehungen durch die Auslagerung in vollwertige Elemente möglicher Gegenstand eines Tailorings – anders als in Form eines Attributs direkt am Modellelement hängende Beziehungen.

Die Beziehungen zwischen Elementen des Pakets *Statik* sind pro Vorgehensbaustein unter dem Element *Beziehungen* gesammelt. Abbildung 4.11 stellt einen kleinen Ausschnitt des V-Modell XT Metamodells rund um das Element *Produkt* dar.

Die Elemente *Disziplin* und *Textbaustein* werden direkt über ein Attribut *DisziplinRef* und *TextbausteinRef* des Produkts referenziert (rechts in Abbildung 4.11).

zur zeitlichen Anordnung der Elemente aus dem Paket *Statik*;

- das Paket *Anpassung* beinhaltet die Modellelemente zur Steuerung des Tailorings (es sind im Wesentlichen die Inhalte dieses Pakets, die in der Fallstudie zum V-Modell XT in Abschnitt 7.3 durch Elemente zum feature-basierten Tailoring ausgetauscht werden);
- das Paket *Konventionsabbildungen* setzt Inhalte des V-Modell XT mit anderen Normen und Standards in Beziehung.

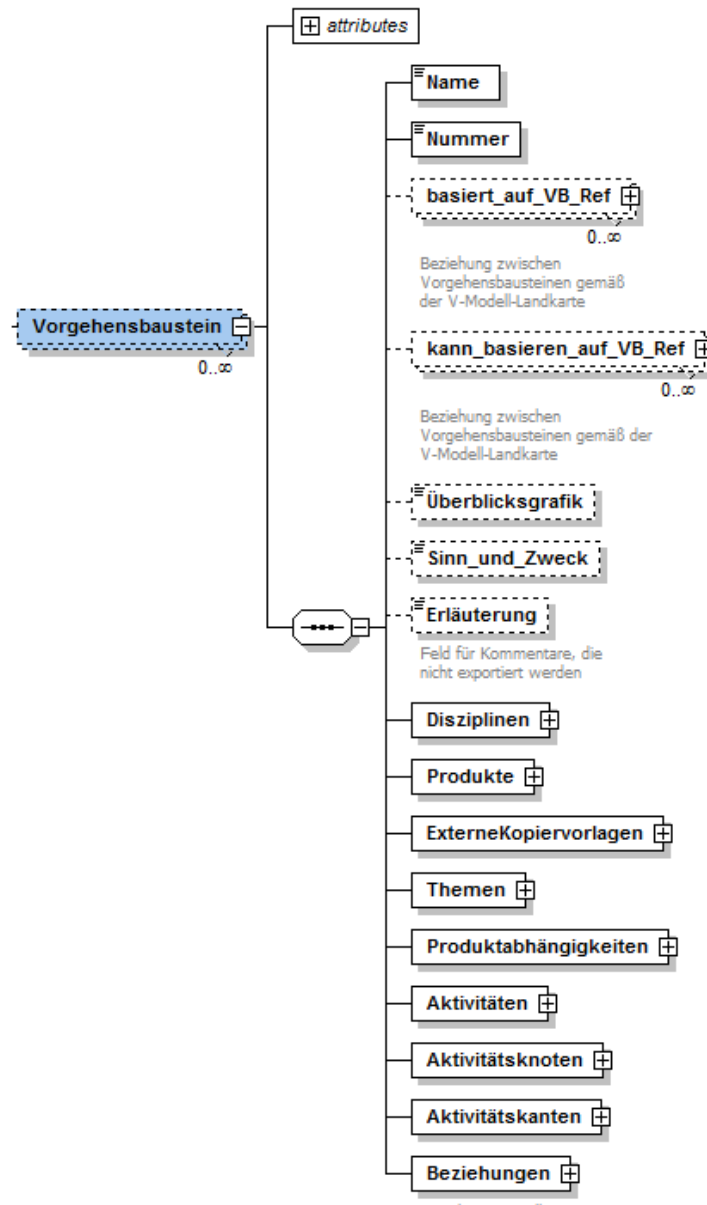


Abbildung 4.9: Struktur eines Vorgehensbausteins

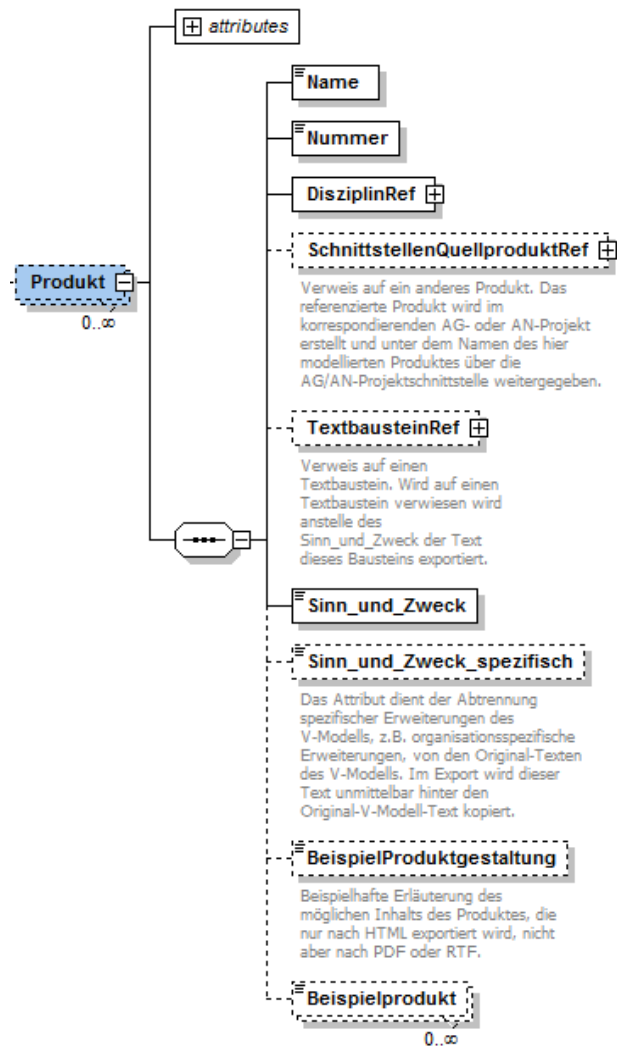


Abbildung 4.10: Struktur eines Produkts

4.2 Analyse: Das V-Modell XT Metamodell

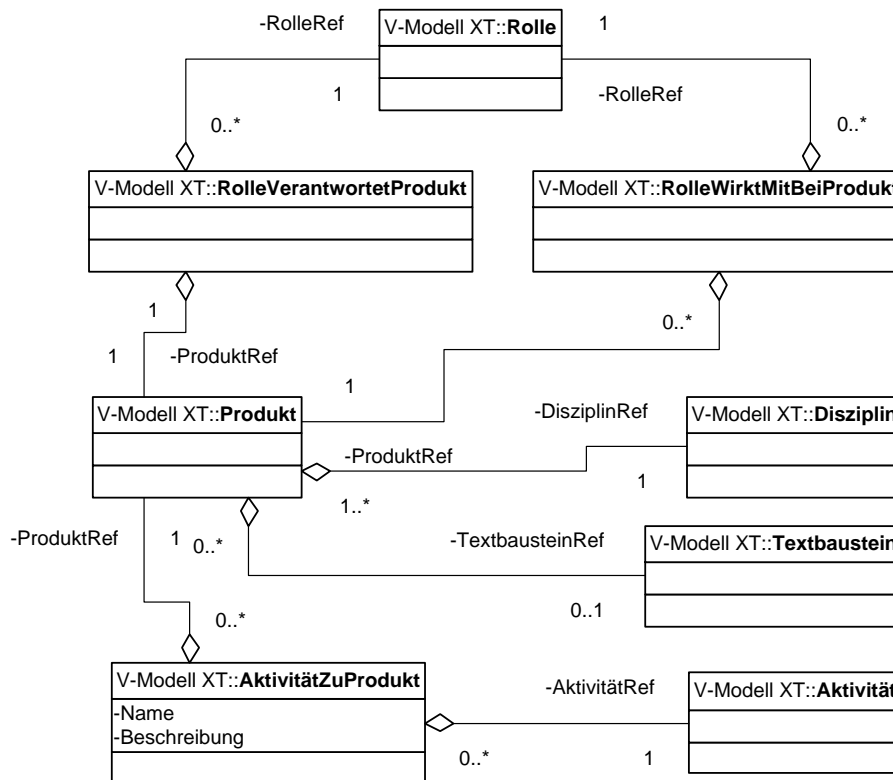


Abbildung 4.11: Beziehungen rund um das Modellelement *Produkt*

Die Verbindung zwischen Produkt und der bearbeitenden *Aktivität* und zwischen Produkt und den beteiligten *Rollen* ist dagegen über *Beziehungselemente* realisiert. Besondere Bedeutung bekommt diese Art der Referenzierung im Fall der Beziehung zwischen Produkt und Rolle (der obere Bereich in Abbildung 4.11). Hier wird über den Typ des Beziehungselements die Art der Beziehung qualifiziert: Die Beziehung *RolleVerantwortetProdukt* hat inhaltlich eine andere Qualität als die Beziehung *RolleWirktMitAnProdukt*.

4.2.3 Fazit

Im vorangegangenen Abschnitt wurde die Struktur des V-Modell XT Metamodells untersucht. Hierzu wurde exemplarisch das Paket *Statik* näher betrachtet. Da die Pakete des V-Modell XT in der Struktur alle gleich aufgebaut sind, lassen sich die Erkenntnisse über das Paket *Statik* auf das gesamte V-Modell XT verallgemeinern: Das V-Modell XT besteht aus einer Menge von Vorgehensmodell-*Elementen*, die miteinander in Beziehung stehen entweder über direkte *Referenzen* oder über *Beziehungselemente*, welche ihrerseits die beteiligten Elemente referenzieren.

Auf abstrakter Ebene stellen Beziehungsknoten lediglich eine Indirektionsstufe zwischen Ausgangs- und Zielelement der Beziehung dar. Das V-Modell XT bildet also einen gerichteten Graph ohne Mehrfachkanten.

Die Knoten des Graphen entsprechen den *referenzierbaren Elementen* des V-Modell XT. Zwar kann ein referenzierbares Element in der XML Repräsentation des V-Modell XT weitere XML Kindelemente haben – diese dienen aber lediglich

der Strukturierung der Inhalte innerhalb des Modellknotens und können selbst nicht an Beziehungen teilnehmen.

Die Kanten des Graphen entsprechen den -Ref Elementen im V-Modell XT. Wie in Abschnitt 4.2.1 ausgeführt, kommen diese Elemente entweder als Attribut eines anderen Modellelements oder als Teil eines Beziehungselements vor.

4.3 Analyse: das Metamodell SPEM

Das Software & Systems Process Engineering Meta-Model (SPEM) [OMG08] ist ein Metamodell zur Definition von Softwareentwicklungsprozessen. SPEM selbst basiert wiederum auf der Meta Object Facility (MOF) [OMG11a].

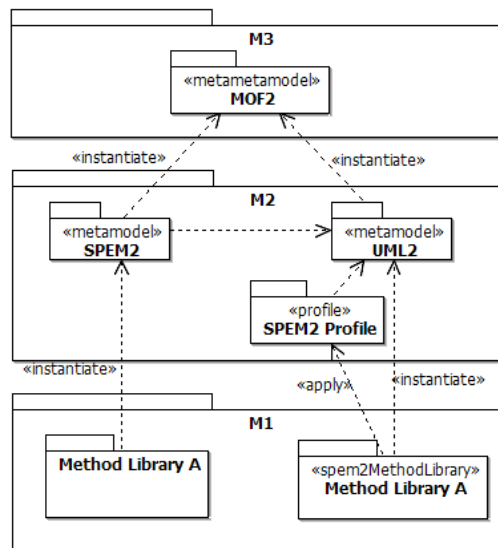


Abbildung 4.12: Modellierungsebenen bei SPEM und UML

Abbildung 4.12 macht die Beziehung zwischen einem konkreten Vorgehensmodell (auf Ebene M1), dem Metamodell SPEM (auf Ebene M2) und dem Metameta-Modell MOF (auf Ebene M3) deutlich.

4.3.1 Paketstruktur

Das Metamodell SPEM ist in eine Paketstruktur zergliedert. Die Pakete bauen dabei aufeinander auf und erweitern oder spezialisieren die Inhalte der eingebundenen Pakete. Abbildung 4.13 zeigt die gesamte Paketstruktur des SPEM Metamodells.

Die Pakete im Einzelnen sind [OMG08]:

Core: In diesem Paket sind die Basisklassen für alle anderen Pakete enthalten. In diesem Paket werden erstens Klassen definiert, welche die Möglichkeit erlauben, *Kinds* von SPEM Klassen zu unterscheiden und zweitens enthält Core eine Reihe von abstrakten Klassen zur Beschreibung von „Arbeit“ (engl. „work“).

Process Structure: Dieses Paket enthält die wesentlichen Klassen zur Beschreibung der statischen Vorgehensmodellstruktur. Hier werden die Strukturen für eine Aktivitätshierarchie definiert (*Work Breakdown Structure*), ebenso

4.3 Analyse: das Metamodell SPEM

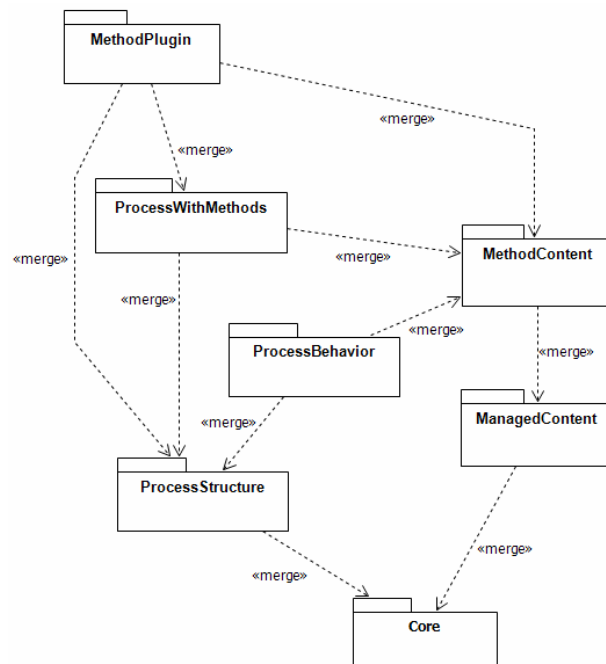


Abbildung 4.13: Paketstruktur des SPEM Metamodells

wie die Beziehungen zwischen diesen Aktivitäten und bearbeitenden Rollen. Weiterhin enthält Process Structure die Elemente zur Beschreibung von Ein- und Ausgabeprodukten (*Work Products*).

Process Behavior: Dieses Paket erweitert die statische Struktur des Vorgehensmodells um die Möglichkeit, „Verhalten“, d.h. zeitliche Ordnung der Aktivitäten oder etwa ein Zustandsmodell für Arbeitsergebnisse zu definieren. Das Paket Process Behavior ist dabei nicht selbst das Metamodell zur Beschreibung von Verhalten, sondern verweist auf *externe* Beschreibungstechniken, beispielsweise Aktivitätsdiagramme oder Zustandsautomaten.

Managed Content: Das Paket Managed Content führt Konzepte zur menschenlesbaren, textuellen Beschreibung von Elementen ein. Diese Elemente bilden den Grundstein für eine *Prozessdokumentation*.

Method Content: Dieses Paket erweitert das Metamodell um die Möglichkeit, die Elemente eines Vorgehensmodells zu Methodenbausteinen zu gruppieren. Ein Methodenbaustein verheiratet die Strukturelemente aus dem Paket *Process Structure* mit den Elementen zur textuellen Beschreibung aus dem Paket *Managed Content* zu einer in sich abgeschlossenen Methodenbeschreibung. Kernelemente einer Methode sind dabei Aktivitätsbeschreibungen, involvierte Rollen und Ein- bzw. Ausgabe-Arbeitsprodukte.

Process With Methods: In diesem Paket wird das Metamodell um Konzepte erweitert, die es erlauben, Methodenbausteinen in den Lebenszyklus eines Prozesses einzuordnen. In diesem Paket werden Konzepte wie Phase und Meilenstein eingeführt.

Method Plugin: Das Method Plugin Paket erweitert das Metamodell um Mechanismen zur Erstellung und Verwaltung von Methodenbibliotheken. Die in diesem Paket eingeführten Konzepte Methoden-Plug-In, Prozesskomponente und Variabilität erlauben die Gestaltung, Modularisierung und Erweiterung eines konkreten Vorgehensmodells ausgehend von vorhande-

nen, in Methodenbibliotheken verwalteten, Inhalten. Zu den Möglichkeiten der Konfiguration von SPEM-basierten Vorgehensmodellen vergleiche auch 3.2.2.

4.3.2 Method Content

Die Kernbestandteile eines Vorgehensmodells sind in SPEM im Paket *Method Content* beschrieben. Tabelle 4.1 bietet einen Überblick über die fünf wichtigsten Klassen im Paket *Method Content*.

Element	Beschreibung
WorkProductDefinition	Dieses Element beschreibt Arbeitsergebnisse, also Dokumente oder sonstige Produkte. Tasks verwenden WorkProducts als Ein- oder Ausgabe und verwenden, verändern oder erzeugen WorkProducts. WorkProducts können zudem untereinander in Beziehung stehen.
RoleDefinition	Mit einer Role wird angegeben, wer einen Task durchführt. Hierzu enthält die Rolle eine Beschreibung des erforderlichen Fähigkeitenprofils.
TaskDefinition	Ein Task beschreibt einen Arbeitsvorgang, welcher von einer oder mehrerer Rollen bearbeitet wird. Ein Task kann in mehrere Unteraufgaben unterteilt sein. Ein Task steht in Beziehung zu einer Reihe von Eingabeprodukten, die verwendet oder modifiziert werden, um eine Reihe von Ausgabeprodukten zu erstellen.
Category	Mit Hilfe des Category Elements ist es möglich, andere Methodenelemente hierarchisch zu strukturieren.
Guidance	Mit dem Guidance Element kann ein bestehendes Element um zusätzliche Informationen erweitert werden. Guidance Elemente tragen üblicherweise Hinweise zur Erstellung, Vorlagen, Checklisten, etc.

Tabelle 4.1: Kernelemente eines SPEM Methodenbausteins

Die in Tabelle 4.1 beschriebenen Klassen erlauben die Definition eines in sich abgeschlossenen Methodenbausteins mit Aufgaben, Verantwortlichkeiten, Ergebnissen und detaillierten Hinweisen und Beschreibungen. Diese Definition erfolgt unabhängig von der tatsächlichen Verwendung des Methodenbausteins im Gesamtprozess.

4.3.3 Zusammenhang von Prozessstruktur und Methoden

Die Einbettung eines Methodenbausteins in die Struktur des im Paket *Process Structure* definierten Gesamtverfahrens erfolgt über die im Paket *Process With Methods* eingeführten Erweiterungen. Das Paket *Process Structure* definiert die

4.3 Analyse: das Metamodell SPEM

(statische) Gesamtstruktur eines Vorgehensmodells unabhängig von der inhaltlichen Ausgestaltung. Da bei der Definition der Gesamtstruktur noch nicht bekannt ist, mit welchen konkreten Inhalten das Vorgehensmodell ausgestaltet wird, ist die Struktur durch Platzhalterklassen beschrieben. Die Namen der Platzhalterklassen in SPEM enden auf *-Use*. Deren Platz wird bei der inhaltlichen Ausgestaltung jeweils von einem konkreten Element, dessen Namen auf *-Definition* endet, eingenommen. Durch die lose Kopplung von Inhalten und Struktur des Gesamtvorgehens wird erreicht, dass

- Methodenbeschreibungen auch unabhängig von ihrer Verwendung im Gesamtprozess eine vollständige und gültige Einheit sind.
- die Struktur des Vorgehensmodells auch unabhängig von der Ausgestaltung mit konkreten Inhalten gültig ist.
- ein und dieselbe Prozessstruktur mit unterschiedlichen konkreten Methodeninhalten ausgestaltet werden kann.
- Methodenelemente in unterschiedlichen Vorgängen oder Phasen des Prozesses unterschiedlich verwendet werden können. Betrachtet man beispielsweise den Prozess in der Küche eines Spezialitätenrestaurants, so wird die Aufgabe „Speise zubereiten“ für die Vorspeise von der Rolle „Potager“ bearbeitet, während die Bearbeitung der entsprechenden Aufgabeninstanz in der Phase „Hauptspeise“ dem „Saucier“ obliegt.

Die den Methodenelementen aus Tabelle 4.1 gegenüberstehenden Platzhalterklassen in *Process Content* sind in Tabelle 4.2 zusammengefasst.

Element	Beschreibung
Activity	Über Activity Elemente können Teilprozesse gruppiert und hierarchisiert werden. Teilprozesse können im Gesamtprozess mehrfach verwendet werden und über <i>Activity Kinds</i> qualifiziert werden. Über Activities werden Konzepte wie Iterationen und Phasen im Prozess abgebildet.
WorkProductUse	Dieses Element dient als Platzhalter für ein in <i>Method Content</i> definiertes Arbeitsergebnis.
RoleUse	Dieses Element dient als Platzhalter für eine in <i>Method Content</i> definierte Rolle.
TaskUse	Dieses Element dient als Platzhalter für einen in <i>Method Content</i> definierten Task.
Guidance	Wie schon bei der Methodenbeschreibung lassen sich über Guidance Elemente zusätzliche Informationen, wie z. B. Checklisten, Vorlagen und Hinweise definieren.

Tabelle 4.2: Kernklassen in SPEM *Process Structure*

Der Zusammenhang zwischen *TaskDefinition*, *RoleDefinition* und *WorkProductDefinition* aus *Method Content* und den Elementen *TaskUse*, *RoleUse* und *WorkProductUse* ist in Abbildung 4.14 dargestellt. Die Verbindung wird dabei über eine Spezialisierung des Elements *BreakdownElement* aus dem Paket *Process Structure* zu *MethodContentUse* realisiert. Ein abstraktes *MethodContentUse* Element in der Prozessstruktur steht also stellvertretend für ein konkretes Element aus *Method Content*.

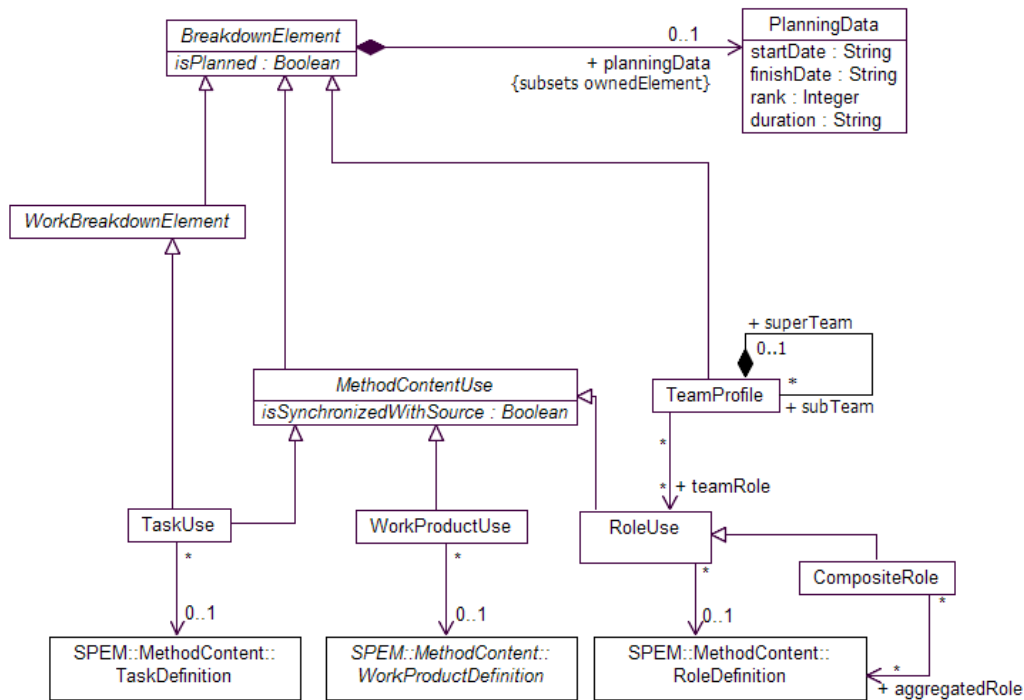


Abbildung 4.14: Zusammenhang zwischen Methodenbausteinen und Prozessstruktur

Wie im unteren Teil von Abbildung 4.14 zu erkennen, wird die Stellvertreterbeziehung durch eine direkte Assoziation zwischen den *-Use* Elementen und den *-Definition* Elementen realisiert. Ein *-Use* Element kann immer nur auf höchstens eine konkrete Methodelement-Definition verweisen, während eine Methodelement-Definition in mehreren Kontexten des Prozesses (unterschiedlich) „benutzt“ werden kann. Die *-Use* Elemente stellen also lediglich eine Indirektionsstufe zwischen der Definition der Prozessstruktur und den Methodeninhalten dar.

4.3.4 Physikalische Repräsentation

Das Metamodell SPEM basiert seinerseits auf dem Meta-Metamodell *Meta Object Facility* (MOF) [OMG11a]. Auf dem MOF-Standard basierende Modelle werden physikalisch typischerweise in Form von XMI Dokumenten dargestellt (XML Metadata Interchange) [OMG11b]. Dieser Standard der Object Management Group (OMG) dient als XML-Austauschformat für Modelle basierend auf MOF. XML-Vokabulare für XMI-Dokumente können in Form von Document Type Definitions oder als XSD Schema erstellt werden.

Da jedes auf MOF basierende Metamodell als XMI-Dokument darstellbar ist, können wir uns für die Betrachtungen hier eine detaillierte Diskussion der physikalischen Repräsentation ersparen und verweisen auf [OMG11a, OMG11b].

Für die Analyse der Struktur betrachten wir die in MOF definierten Metaklassen näher. Wie auch das darauf aufbauende Metamodell SPEM ist die Meta Object Facility in eine Paketstruktur zergliedert. Die die MOF bildenden Pakete sind:

Common Dieses Paket enthält mit *ReflectiveCollection* und *ReflectiveSequence* zwei Klassen, die in den anderen Paketen der MOF verwendet werden.

4.3 Analyse: das Metamodell SPEM

Reflection Dieses Paket enthält alle Metamodellkonstrukte, die zur Selbstbeschreibung eines auf MOF basierenden (Meta-)Modells nötig sind. Durch diese Konstrukte ist es möglich, über die Beschaffenheit eines auf MOF basierenden (Meta-)Modells zu reflektieren.

Identifiers Dieses Paket enthält Konstrukte, die eine eindeutige Identifikation von (Meta-)Modellelementen erlauben.

Extension Dieses Paket enthält Konstrukte zur Erweiterung von (Meta-)Modellelementen durch Metadaten. Die Metadaten werden hierzu in Name-Wert-Paaren zum betreffenden Element hinterlegt. Dieses Konstrukt erlaubt insbesondere Werkzeugherstellern, (Meta-)Modelle durch herstellerspezifische Informationen anzureichern.

Abbildung 4.15 enthält einen Überblick über alle MOF bildenden Pakete. EMOF in der Abbildung steht für Essential MOF und CMOF für Complete MOF. Complete MOF erweitert gegenüber Essential MOF die Fähigkeit des (Meta-)Modells zur Selbstbeschreibung.

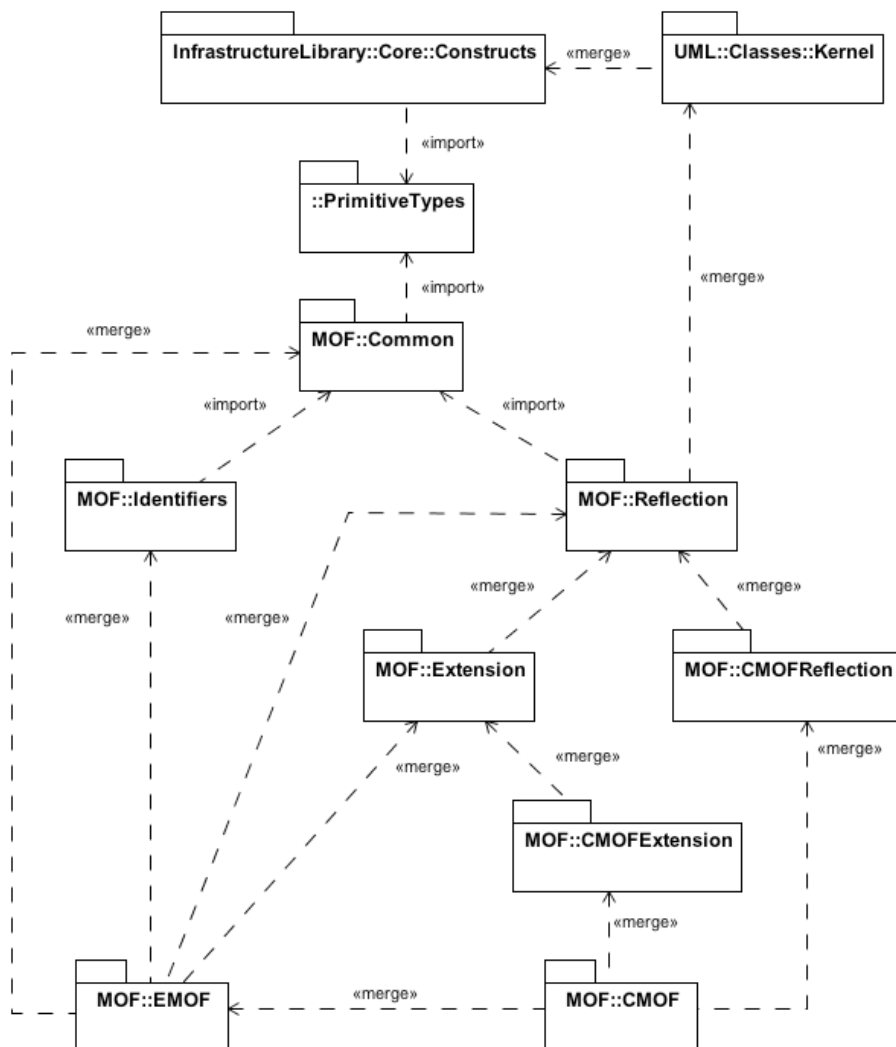


Abbildung 4.15: Überblick über die Pakete in MOF [OMG11a]

Entscheidend sind die oben in Abbildung 4.15 dargestellten Pakete `InfrastructureLibrary::Core::Constructs` und `UML::Classes::Kernel`. Diese wer-

den aus der *UML Infrastructure Library* [OMG11c] importiert. Das heißt insbesondere, dass die Bildung der Struktur eines MOF (Meta-Modells) mit Hilfe der Konstrukte der *UML Infrastructure* erfolgt. Beziehungen zwischen (Meta-) Modellelementen in MOF und damit in SPEM werden durch das Zusammenspiel der *UML Infrastructure*-Klassen *Class*, *Property* und *Association* gebildet.

4.3.5 Fazit

Im vorangegangenen Abschnitt haben wir das Metamodell SPEM untersucht. Dieses stützt sich auf das Meta-Metamodell MOF [OMG11a] ab, welches seinerseits Konstrukte aus der *UML Infrastructure* einbindet [OMG11c]. Der modulare Aufbau und das Zusammenspiel der Klassen auf verschiedenen Metaebenen macht das SPEM Metamodell sehr komplex. Letztendlich handelt es sich aber um ein Metamodell, welches mit einfachen UML-Mitteln beschrieben ist: Elemente im Metamodell bauen auf dem UML-Konstrukt *Klasse* auf und Beziehungen zwischen Elementen werden durch das UML-Konstrukt *Association* dargestellt. Ein mit den Mitteln von SPEM beschriebenes Vorgehensmodell ist daher ein – wenn auch komplexer – Graph.

4.4 Analyse: Formalisierung leichtgewichtiger Methoden

In Abschnitt 3.3 und insbesondere in Tabelle 3.3 haben wir die Kerninhalte einiger populärer agiler Vorgehensweisen dargestellt. Ob und inwieweit sich diese Inhalte formalisiert darstellen lassen und ob sie Angriffsfläche für Variabilität bieten, diskutieren wir im folgenden Abschnitt.

Feature Driven Development

Abbildung 4.16 stellt den Prozess und das Datenmodell des Feature Driven Development dar.

Das Diagramm in Abbildung 4.16 sieht zwar auf den ersten Blick umfangreich aus, bei näherer Betrachtung lassen sich aber weder im Bereich des Prozesses auf der linken Seite noch im Bereich des Datenmodells auf der rechten Seite sinnvolle Variabilitätspunkte identifizieren. Der Prozess ist so minimal, dass er keinen Spielraum für projektspezifische Anpassung zu Projektbeginn erlaubt. Die Artefakte auf der rechten Seite des Diagramms sind eine direkte Folge der Aktivitäten auf der linken. Auch hier gibt es keine sinnvollen Ansatzpunkte für projektspezifische Anpassung im Sinne der Konfiguration des Modells.

Extreme Programming

Zu Extreme Programming gibt es keine Erfahrungsberichte, die von der Anwendung aller 24 Praktiken in einem Projekt berichten [ASRW02, Gre01, Sch01]. Diese Beobachtung könnte nahelegen, dass die Praktiken in Extreme Programming ein geeigneter Kandidat für projektspezifische Anpassung seien. Ob eine Praktik im Projekt Anwendung findet, hängt allerdings nicht davon ab, ob sie in der Prozessdokumentation steht, sondern ob sie Eingang in die Kultur des Teams findet und *lebt wird*. Anstatt der projektspezifischen Konfiguration von Praktiken zu Projektbeginn erscheinen hier Ansätze wie [MKRS⁺08] sinnvoller.

4.4 Analyse: Formalisierung leichtgewichtiger Methoden

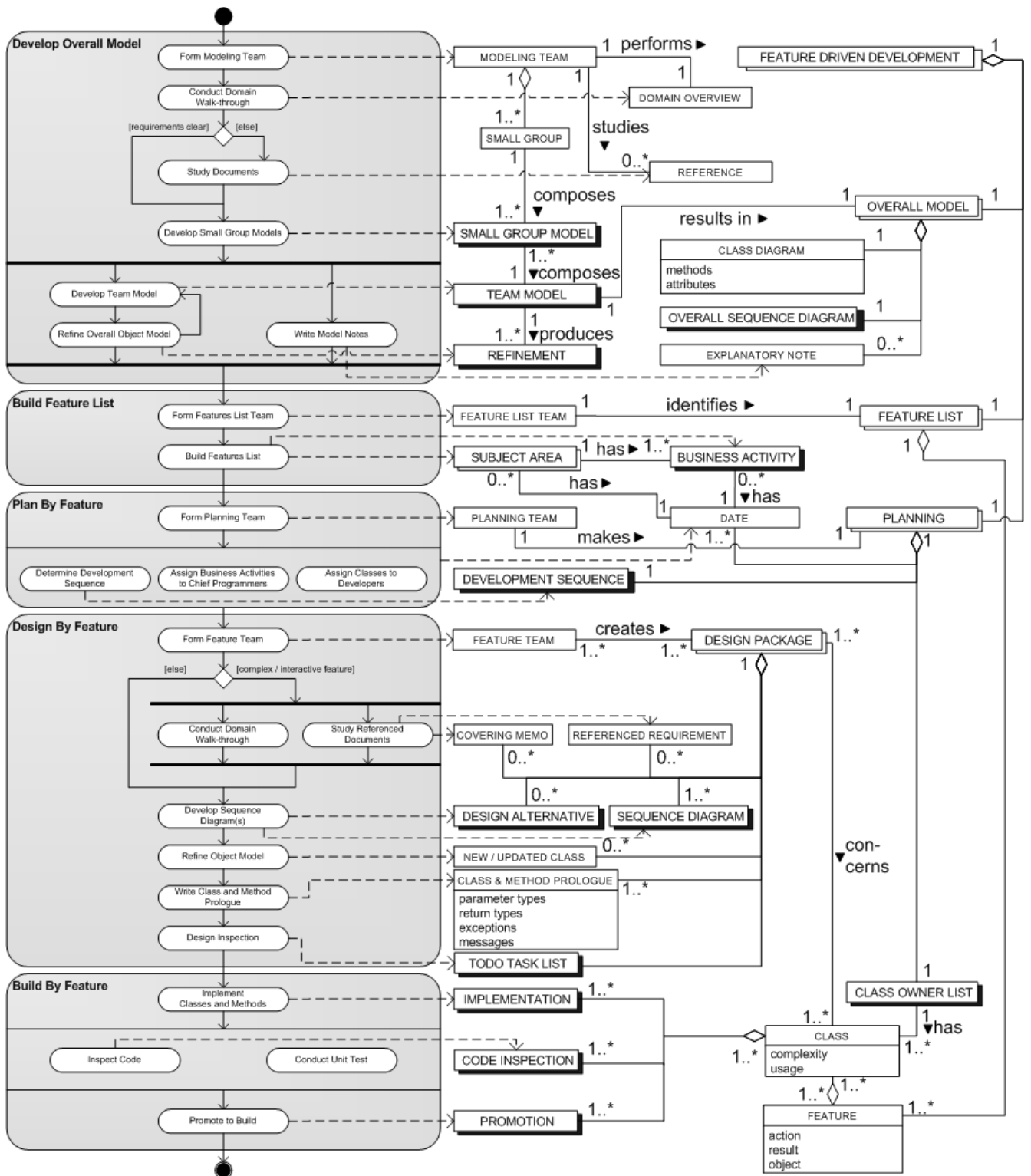


Abbildung 4.16: Prozess-Daten Diagramm für Feature Driven Development [vG12]

Scrum

Für Scrum gibt es Werkzeugumsetzungen – beispielsweise mit der MSF for Agile Prozessvorlage [Mic12b] und dem Visual Studio Scrum [Mic12c] für den Team Foundation Server [DGG06] – zur Unterstützung der wichtigsten Scrum-Konzepte. Diese stellen eine Formalisierung der Konzepte von Scrum dar. Sie sind allerdings so minimal (Visual Studio Scrum definiert 5 *Work Item Types* und bietet Werkzeugunterstützung zur Anzeige von Burndown Charts) und generisch, dass sich diese nicht sinnvoll im Sinne einer Konfiguration des Modells projektspezifisch anpassen lassen.

4.4.1 Fazit

Prinzipiell lassen sich auch leichtgewichtige Methoden formalisieren und beispielsweise in einer Graphstruktur darstellen. Da derartige Vorgehensweisen aber möglichst minimal gehalten sind, gibt es keine sinnvollen Ansatzpunkte für projektspezifische Anpassung im Sinne einer Konfiguration eines Modells.

Wie in Kapitel 3 ausgeführt, unterliegen natürlich auch leichtgewichtige Methoden der projektspezifischen Anpassung und Interpretation. Allerdings nicht im Sinne einer Konfiguration des zugrundeliegenden *Modells* (sofern überhaupt vorhanden). Tailoring einer leichtgewichtigen Methode bedeutet in der Regel eine Auswahl der Praktiken, die in die Kultur des Teams übernommen werden. Für diese Art der Anpassung eignen sich Verfahren wie z. B. [MKRS⁺08] besser.

Die Formalisierung agiler Praktiken kann dann sinnvoll sein, wenn agile Praktiken als Teil eines „größeren“, reichhaltigen Vorgehensmodells in eine projektspezifische Vorgehensmodellinstanz aufgenommen werden sollen. In Kapitel 1 haben wir angedeutet, dass eine Kombination reichhaltiger Vorgaben mit agilen Praktiken sinnvoll sein kann, beispielsweise wenn sich das Gesamtprojektteam über mehrere Standorte verteilt (Bedarf für formale Dokumentation, Kommunikation und definierte Schnittstellen [KK08]), die Einzelteams aber so unabhängig voneinander sind, dass sie im „Mikrokosmos“ jeweils einer leichtgewichtigen Vorgehensweise folgen können.

Im V-Modell XT wird mit der Entwicklungsstrategie *prototypische Entwicklung* eine „agile“ Entwicklungsstrategie für die Systementwicklung unterstützt (auch wenn sie nicht so bezeichnet wird). In der Dokumentation [RHB⁺06] zu dieser Entwicklungsstrategie heißt es: „Typisch für diese Entwicklungsstrategie ist darüber hinaus die Präsenz des Auftraggebers auf der Auftragnehmerseite während der Entwicklung“. Dies entspricht genau der aus Extreme Programming bekannten Vorgabe der Anwesenheit eines *On Site Customer*. Für derartige Fälle mag es sinnvoll sein, entsprechende agile Praktiken mit in die Dokumentation des Vorgehensmodells aufzunehmen.

Der OpenUP [Ecl11b] ist ein weiteres Beispiel für die Integration agiler Praktiken in ein formalisiertes Modell.

4.5 Ein einfaches Meta-Metamodell für Vorgehensmodelle

Ausgehend von den Beobachtungen über die innere Struktur von Vorgehensmodellen entwickeln wir im Folgenden ein minimales Meta-Metamodell für

4.5 Ein einfaches Meta-Metamodell für Vorgehensmodelle

Vorgehensmodelle. Dieses stellt als solches keinen eigenen Beitrag der vorliegenden Arbeit dar, sondern dient lediglich der Abstraktion von konkreten Vorgehensmodell-Metamodellen bei der Entwicklung und Illustration der Konzepte und Techniken zum feature-basierten Tailoring. Das einfache Meta-Metamodell dient uns somit der Vorgehensmodell-unabhängigen Darstellung der wesentlichen Konzepte. Wie die Umsetzung in einem konkreten Vorgehensmodell aussehen kann, beschreiben wir in Kapitel 6 und führen diese exemplarisch in den Fallstudien in Kapitel 7 durch.

Grundlegende Struktur

Dem Meta-Metamodell liegt folgende Überlegung zugrunde: ein Vorgehensmodell besteht aus beliebigen, strukturierten und miteinander in Beziehung stehenden Inhalten. Inhaltselemente bilden durch hierarchische Komposition eine logische Einheit. Beziehungen im Modell sollen nur möglich sein zwischen derartigen logischen Einheiten. Es ergibt sich das in Abbildung 4.17 dargestellte Basismodell:

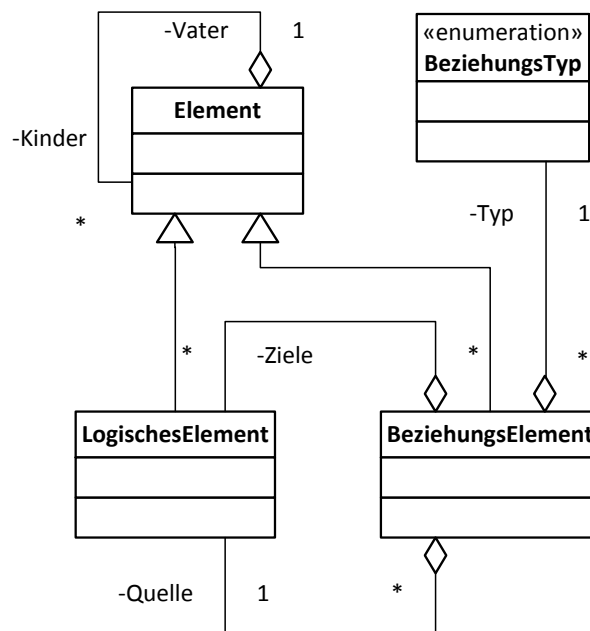


Abbildung 4.17: Kern des einfachen Meta-Metamodells

Element Das Element ist der zentrale *Inhaltsträger* im Modell. Wir nehmen an, dass es, für die folgende Betrachtung nicht näher spezifizierte „Inhalte“, beispielsweise einen Namen und eine Beschreibung oder ein Diagramm tragen kann. Das Element kann selbst weitere Elemente aggregieren und ermöglicht dadurch eine hierarchische Strukturierung von Inhalten.

LogischesElement Ein logisches Element ist ein *Element*, welches an Beziehungen zu anderen *LogischesElementen* teilnehmen kann. Durch das logische Element ermöglichen wir die Bildung von logisch zusammengehörigen Inhaltseinheiten, die ihrerseits eine interne hierarchische Struktur haben können.

Beziehungselement Das *Beziehungselement* stellt eine gerichtete Kante zwischen

einem *LogischesElement* als Quelle der Beziehung und mehreren *LogischesElementen* als Ziel der Beziehung dar. Das *BeziehungsElement* stellt also eine 1 : *m* Beziehung zwischen logischen Elementen her. Eine Beziehung ist durch einen *BeziehungsTyp* qualifiziert. Alternativ könnte man die Typisierung der Beziehungen auch durch Bildung von Unterklassen von *BeziehungsElement* realisieren. Da ein *BeziehungsElement* selbst ein *Element* ist, kann es Inhalte, beispielsweise eine Beschreibung der Beziehung, beinhalten.

BeziehungsTyp Über einen *BeziehungsTyp* wird ein *BeziehungsElement* qualifiziert.

Variabilitätspunkte

Würde man beim projektspezifischen Tailoring Variabilität über das ganze Modell erlauben, könnten durch das Tailoring inhaltlich und logisch zusammengehörige Einheiten aufgebrochen werden und dadurch eine inhaltlich nicht sinnvolle projektspezifische Instanz des Vorgehensmodells entstehen. Wir führen deshalb zur Darstellung der Variabilität im Vorgehensmodell ein *VariablesElement* ein (siehe Abbildung 4.18). Nur als variabel gekennzeichnete logische Einheiten im Vorgehensmodell sollen Gegenstand des Tailorings sein können. Wir gehen auf die Bedeutung und Semantik der Variabilität im folgenden Abschnitt 4.5 ein.

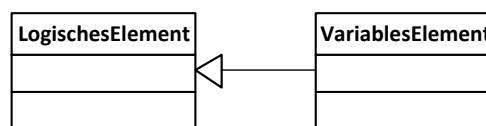


Abbildung 4.18: Variabilitätspunkte im Meta-Metamodell

In der folgenden Darstellung verwenden wir folgende Mengen:

Elemente := die Menge aller Objekte vom Meta-Typ *Element*

LogischeElemente := die Menge aller Objekte
vom Meta-Typ *LogischesElement*

BeziehungsElemente := die Menge aller Objekte
vom Meta-Typ *BeziehungsElement*

VariableElemente := die Menge aller Objekte
vom Meta-Typ *VariablesElement*

Wie den Abbildungen 4.17 und 4.18 zu entnehmen, sind *LogischesElement* und *BeziehungsElement* eine Spezialisierung von *Element*. *VariablesElement* erbt weiterhin von *LogischesElement*. Damit ergibt sich für ein Vorgehensmodell als Instanz des Meta-Metamodells:

4.5 Ein einfaches Meta-Metamodell für Vorgehensmodelle

$$\text{LogischeElemente} \subset \text{Elemente}$$

$$\text{BeziehungsElemente} \subset \text{Elemente}$$

$$\text{VariableElemente} \subset \text{LogischeElemente}$$

Für die *BeziehungsElemente* fordern wir, dass beide Enden der Beziehung (Quelle und Ziele) im Modell vorhanden sind. Würde ein *BeziehungsElement* durch Entfernen eines der beiden teilnehmenden *LogischesElement* zu einem „dangling Pointer“, so wäre das *BeziehungsElement* ebenfalls zu entfernen. Wir fordern also:

$$\begin{aligned} \forall b \in \text{BeziehungsElemente} \mid \\ \exists q \in \text{LogischeElemente} : b.\text{Quelle} = q \quad \wedge \\ b.\text{Ziele} \neq \emptyset \end{aligned}$$

Unter der Annahme, dass alle Inhalte eines Vorgehensmodells über entsprechende *Elemente* und Beziehungen zwischen diesen darstellbar sind, kann das Vorgehensmodell aufgefasst werden als die Menge aller Elemente:

$$\text{Vorgehensmodell} := \text{Elemente}$$

Semantik der Variabilität

Im Sinne der Produktlinienmodellierung (vergleiche Abschnitt 2.6) fassen wir ein projektspezifisches Vorgehensmodell auf als ein *Produkt* der durch ein Konfigurationsmodell definierten Produktlinie auf. Mit diesem Verständnis bedeutet das projektspezifische Tailoring die Entfernung der in der projektspezifischen Instanz nicht benötigten Modellanteile.

Seien $ve \in \text{VariableElemente}$ ein variables Element, welches im Zuge der Konfiguration aus der projektspezifischen Instanz zu entfernen ist. Sei weiterhin $\text{Teilbaum}(x)$, $x \in \text{Elemente}$ die Menge aller Elemente im durch die Eltern-Kind-Beziehung zwischen *Elementen* aufgespannten Baum mit der Wurzel x (vergleiche Abbildung 4.17). Damit ist die Gesamtmenge aller aus der projektspezifischen Instanz zu entfernenden Elemente:

$$\begin{aligned} \text{ZuEntfernendeElemente}(ve) := \{x : \\ x = ve \quad \vee \\ x \in \text{Teilbaum}(ve)\} \end{aligned}$$

Da wir ein variables Element mit seiner hierarchischen inneren Struktur als eine logische Einheit auffassen, werden alle Kindelemente eines zur Entfernung markierten variablen Elements ebenfalls entfernt. Damit enthält die Menge der zu entfernenden Elemente den kompletten Teilbaum unterhalb des zur Entfernung markierten variablen Elements und das variable Element selbst.

Zudem werden alle *BeziehungsElemente*, von denen entweder das Quell-Element oder alle Ziel-Elemente durch das Tailoring entfernt werden sollen, ebenfalls entfernt. Ein *BeziehungsElement* bleibt so lange im Modell erhalten, wie es ein

Quell-Element und mindestens ein Ziel-Element hat. Die Menge der zu entfernenden *BeziehungsElemente* ist daher:

$$\begin{aligned} \text{ZuEntfernendeBeziehungen}(ve) := \{b : \\ b \in \text{Beziehungselemente} \wedge \\ b.\text{Quelle} = ve \vee \\ b.\text{Ziele} \setminus \{ve\} = \emptyset\} \end{aligned}$$

Das Ergebnis des Tailorings entspricht damit:

$$\begin{aligned} \text{Tailoring}(ve) := \text{Vorgehensmodell} \setminus \\ \text{ZuEntfernendeElemente}(ve) \cup \text{ZuEntfernendeBeziehungen}(ve) \end{aligned}$$

Das einfache Meta-Metamodell mit der vereinbarten Semantik bildet die Grundlage für das feature-basierte Tailoring in Kapitel 5. Die als *VariablesElement* ausgezeichneten Elemente des Vorgehensmodells bilden dabei den „Angriffspunkt“ für die Konfiguration und stellen damit (über die Tailoring-Operationen – siehe Abschnitt 5.3) die Verbindung zwischen Vorgehensmodellinhalten und Konfigurationsmodell dar.

Zunächst überprüfen wir die Eignung des Meta-Metamodells, indem wir es anhand des V-Modell XT Metamodells beispielhaft instanziiieren.

Beispielhafte Instanziierung: V-Modell XT

Im folgenden Abschnitt weisen wir nach, dass die Ausdrucksmächtigkeit des einfachen Meta-Metamodells für die Entwicklung des feature-basierten Tailorings ausreicht. Wir führen den Nachweis durch, indem wir die Instanzierungsregeln für das V-Modell XT Metamodell angeben. Aufgrund des Umfangs beschränken wir uns hierbei auf den „Teilbaum“ unterhalb eines Vorgehensbausteins. Tabelle 4.3 stellt für jedes V-Modell XT Metamodellelement die Meta-Metaklasse aus Abschnitt 4.5 dar.

Die Abbildung 4.19 illustriert den Zusammenhang zwischen Meta-Metamodell (M3), Metamodell (M2) und (Vorgehens-)Modell (M1) anhand des Zusammenspiels der Metamodellelemente *ErzeugendeAbhängigkeit*, *ErzeugendeAbhängigkeitVonProdukt* und *ErzeugendeAbhängigkeitZuProdukt*, der entsprechenden Meta-Metamodellelemente und – unten in der Abbildung dargestellt – durch Modellelemente als Instanzen der Metamodellelemente (vergleiche hierzu auch Abschnitt 2.2.1).

Einige Bemerkungen zur Tabelle 4.3:

- Die Metamodellelemente *Vorgehensbausteine*, *Disziplinen*, *Produkte*, *Themen*, *Produktabhängigkeiten*, *Aktivitäten*, *Aktivitätsknoten*, *Aktivitätskanten* und *Beziehungen* sind ein Beispiel für Elemente, die rein der hierarchischen Gruppierung dienen. Sie tragen selbst keinen Inhalt (außer ihre Kinder) und können an keinen Beziehungen teilnehmen.
- Die Metamodellelemente *ProduktZuDisziplin* und *ArbeitsschrittknotenZuAktivität* finden sich nicht im V-Modell XT Metamodell. Die Beziehungen zwischen *Produkt* und *Disziplin* bzw. zwischen *Arbeitsschrittknoten* und

4.5 Ein einfaches Meta-Metamodell für Vorgehensmodelle

V-Modell XT Metamodellelement	Instanz von
V-Modell	Element
V-Modellvariante	Element
Vorgehensbausteine, Disziplinen, Produkte, ExterneKopiervorlagen, Themen, Aktivitäten, Aktivitätsknoten, Aktivitätskanten	Element
Vorgehensbaustein	VariablesElement
Disziplin, Produkt, ExterneKopiervorlage, Thema, Aktivität, Arbeitsschrittknoten	LogischesElement
ProduktZuDisziplin	BeziehungsElement
ArbeitsschrittknotenZuAktivität	BeziehungsElement
Unterthema	Element
Produktabhängigkeiten	Element
ErzeugendeAbhängigkeiten, InhaltlicheAbhängigkeiten, TailoringAbhängigkeiten	Element
ErzeugendeAbhängigkeit, InhaltlicheAbhängigkeit, TailoringAbhängigkeit	LogischesElement
Aktivitätskante	BeziehungsElement
Beziehungen	Element
RollenBeziehungen, Produktabhängigkeitsbeziehungen, Aktivitätsbeziehungen, Produktbeziehungen	Element
RolleVerantwortetProdukt, RolleWirktMitBeiProdukt	BeziehungsElement
TailoringAbhängigkeitVonProdukt, TailoringAbhängigkeitZuProdukt, InhaltlicheAbhängigkeitZuProdukt, ErzeugendeAbhängigkeitVonProdukt, ErzeugendeAbhängigkeitZuProdukt, StrukturAbhängigkeitTeilProdukt, StrukturAbhängigkeitGanzesProdukt	BeziehungsElement
AktivitätZuMethodenreferenz, AktivitätZuWerkzeugreferenz, ArbeitsschrittknotenZuMethodenreferenz, ArbeitsschrittknotenZuWerkzeugreferenz, ArbeitsschrittknotenZuThema, AktivitätZuProdukt	BeziehungsElement
ProduktZuEntscheidungspunkt, ThemaZuProdukt, ExterneKopiervorlageZuProdukt	BeziehungsElement

Tabelle 4.3: V-Modell XT Metamodell als Instanz des Meta-Metamodells

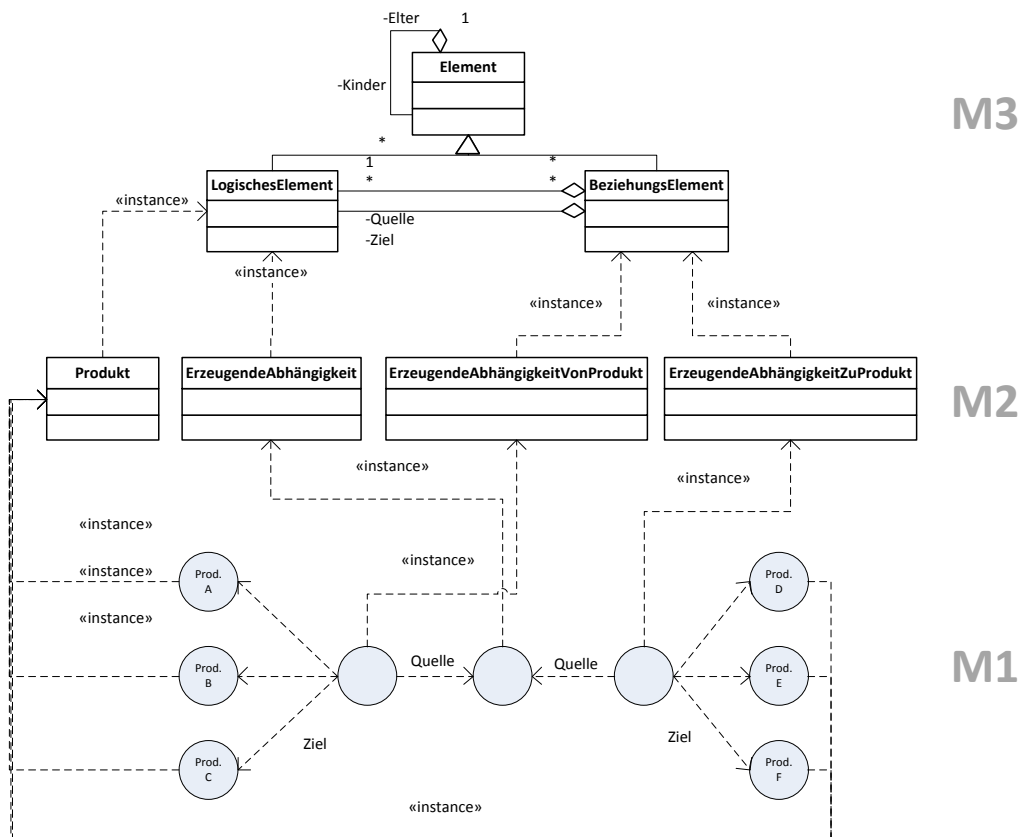


Abbildung 4.19: $n : m$ Beziehung zwischen Produkten

4.6 Tailoring neu formuliert

Aktivität sind im V-Modell XT Metamodell über eine Beziehung als Attribut realisiert (vergleiche hierzu Abschnitt 4.2.1). Die Bedeutung des (Meta-)Modells ändert sich aber nicht, wenn man diese Beziehungen über Beziehungselemente darstellt.

- Die Metamodellelemente *ErzeugendeAbhängigkeit*, *ErzeugendeAbhängigkeitVonProdukt* und *ErzeugendeAbhängigkeitZuProdukt* sind ein Beispiel für die Bildung einer $n : m$ Beziehung – in diesem Fall zwischen Metamodellelementen vom Typ *Produkt*. Abbildung 4.19 illustriert diese Art der Beziehung: die Produkte *Prod. A*, *Prod. B* und *Prod. C* stehen in einer erzeugenden Beziehung mit den Produkten *Prod. D*, *Prod. E* und *Prod. F*. Die Richtung der erzeugenden Beziehung wird dabei über die Typen *ErzeugendeAbhängigkeitVonProdukt* und *ErzeugendeAbhängigkeitZuProdukt* der Beziehungselement-Instanzen ausgedrückt.
- Das Metamodellelement *Vorgehensbaustein* ist die einzige Instanz eines *VariablesElement*. Damit kann nur ein Vorgehensbaustein – und damit immer alle Elemente im Teilbaum unterhalb des Vorgehensbausteins – Gegenstand des projektspezifischen Tailorings sein.

Der vorangegangene Abschnitt hat gezeigt, dass das V-Modell XT Metamodell mit den Mitteln des einfachen Meta-Metamodells beschrieben werden kann. Mit dem *Vorgehensbaustein* als einzigem variablen Element und der in Abschnitt 4.5 vereinbarten Semantik der Variabilität entspricht diese genau der des Tailorings im V-Modell XT: Wird ein Vorgehensbaustein bei der projektspezifischen Anpassung nicht gewählt, so wird das entsprechende Modellelement und alle seine Kinder aus dem Modell entfernt. Existieren zudem Beziehungen in anderen Teilen des Modells (beispielsweise in anderen, angewählten Vorgehensbausteinen), welche auf Modellelemente im entfernten Teilbaum verweisen (bzw. verwiesen haben), so werden diese ebenfalls entfernt.

Da das Meta-Metamodell kein umfassendes Meta-Metamodell für beliebige Vorgehensmodell-Metamodelle sein soll, sondern lediglich als Vehikel zur Entwicklung des feature-basierten Tailorings dient³, genügt uns der Nachweis der Eignung anhand der beispielhaften Abbildung des V-Modell XT Metamodells.

4.6 Tailoring neu formuliert

Aufgabe des Tailorings ist die Wahl passender projektspezifischer Inhalte aus der Gesamtmenge verfügbarer Vorgehensmodellinhalte.

Sieht man von der Möglichkeit der Neuerstellung von Methoden ab, so erfolgt diese Wahl beim *Situational Method Engineering* durch *Hinzunahme* geeigneter Methodenbausteine aus der Methodenbibliothek zu einer Konfiguration (vergleiche Abschnitt 2.5.1).

Beim *Tailoring* im Sinne des V-Modell XT erfolgt die Wahl, indem durch Konfiguration nicht benötigte Inhalte aus der Gesamtmenge der Bausteine *entfernt werden* – die Neuerstellung von Inhalten erfolgt hier typischerweise im Rahmen der organisationsspezifischen Anpassung (vergleiche Abschnitte 2.2.2, 2.5.3 und 3.2.1).

³ Das hier definierte Meta-Metamodell soll in keiner Weise ein Ersatz für weitaus umfangreichere Meta-Metamodelle wie MOF [OMG11a] sein. Es dient lediglich der Veranschaulichung der wesentlichen Konzepte dieser Arbeit – die Verwendung von MOF hätte zu unnötiger Komplexität in der Darstellung geführt.

Für das Ergebnis ist es letztlich unerheblich, ob die projektspezifische Vorgehensmodellinstanz durch *additive* oder *subtraktive* Konfiguration zustande kommt. In Kapitel 5 entwickeln wir Tailoring-Operationen, die sowohl den Einschluss als auch den Ausschluss von Inhalten aus der projektspezifischen Vorgehensmodellinstanz erlauben.

Mit dem einfachen Meta-Metamodell für Vorgehensmodelle aus Abschnitt 4.5 als Grundlage besteht die Aufgabe des Tailorings nunmehr in der Vereinbarung einer für den Prozessingenieur mächtigen und für den Vorgehensmodellanwender möglichst einfach durchzuführenden Konfiguration. Dieser Aufgabe widmen wir uns im folgenden Kapitel 5.

4.7 Zusammenfassung

In diesem Kapitel haben wir uns mit der inneren Struktur von Vorgehensmodellen auseinandergesetzt. Wir haben das V-Modell XT Metamodell und das Software & Systems Process Engineering Meta-Model (SPEM) eingehend analysiert. Aufbauend auf dieser Analyse haben wir in Abschnitt 4.5 ein einfaches Meta-Metamodell für Vorgehensmodelle entwickelt, welches Variabilitätspunkte für die projektspezifische Anpassung vorsieht. Die Eignung des Meta-Metamodells haben wir durch eine Abbildung des V-Modell XT Metamodells auf das Meta-Metamodell nachgewiesen.

In Abschnitt 4.4 haben wir einen Abstecher zu leichtgewichtigen Vorgehensweisen gemacht. Diese lassen sich zwar theoretisch auch in eine formalisierte Struktur überführen, allerdings würde ein derartiger Schritt in den meisten Fällen die Philosophie der entsprechenden Vorgehensweise ad absurdum führen. Zudem wäre aufgrund des minimalen Inhalts leichtgewichtiger Vorgehensweisen der Sinn einer Unterlegung mit einem formalen Metamodell zum Zwecke der projektspezifischen Anpassung mit Hilfe des feature-basierten Tailorings zumindest fragwürdig. Dieser Umstand und die Überlegungen aus Abschnitt 3.3 führen dazu, dass wir agile Methoden und Praktiken im weiteren Verlauf der Arbeit nur mehr entweder als methodischen Bestandteil reichhaltiger Vorgehensmodelle oder als Alternative⁴ betrachten.

⁴ Eine geeignete Projektsituation – insbesondere ein kleines Team – vorausgesetzt.

Feature-basiertes Tailoring

In den vorangegangenen Kapiteln 3 und 4 haben wir die Anpassungsfähigkeit und die dabei verwendete innere Struktur von Vorgehensmodellen analysiert. Wir haben festgestellt, dass die Möglichkeiten zur projektspezifischen Anpassung in populären Vorgehensmodellen entweder beschränkt in ihrer Ausdrucksmächtigkeit oder sehr aufwändig sind. Zudem ist in allen Anpassungsverfahren zwar die Rede von der Einbeziehung der Situationsfaktoren des Projekts – es bleibt aber oft dem Anwender des Vorgehensmodells selbst überlassen, die Charakterisierung des Projekts systematisch vorzunehmen und in der Anpassung geeignet zu berücksichtigen.

In diesem Kapitel entwickeln wir den Ansatz des feature-basierten Tailorings. Das feature-basierte Tailoring umfasst ein Konfigurationsmodell zur systematischen Erfassung der Situation eines Projekts und definierte Operationen, mit welchen die Wirkung der Konfiguration auf die Inhalte des Vorgehensmodells beschrieben wird.

Am Ende dieses Kapitels wird der Leser mit der Theorie des feature-basierten Tailorings vertraut sein und die Kenntnisse und Mittel haben, die Theorie – zusammen mit den methodischen Empfehlungen aus Kapitel 6 – praktisch anzuwenden.

Übersicht

5.1	Überblick	116
5.2	Charakterisierung eines Projekts	117
5.3	Operationen des Tailorings	124
5.4	Feature-basiertes Tailoring von Vorgehensmodellen . .	128
5.5	Sicherstellung von Konsistenz der projektspezifischen Ausprägung	133
5.6	Anwenderunterstützung bei der Durchführung des Tailorings	138
5.7	Zusammenfassung	143

5.1 Überblick

Abbildung 5.1 stellt schematisch den Ansatz des feature-basierten Tailorings dar.

Im unteren Teil der Abbildung sind die Inhalte eines Vorgehensmodells mit Beziehungen untereinander angedeutet. Um in der Darstellung des feature-basierten Tailorings unabhängig von einem konkreten Vorgehens-(Meta-)Modell zu bleiben, verwenden wir in diesem Kapitel zur Darstellung der Inhalte das einfache Meta-Metamodell für Vorgehensmodelle aus Kapitel 4.

Im oberen Teil der Abbildung ist ein Konfigurationsmodell angedeutet, welches zur Charakterisierung der Projektsituation dient.

Verbunden werden die Inhalte und die Projektcharakterisierung in Form des Konfigurationsmodell über festgelegte Operationen auf den Inhalten des Vorgehensmodells.

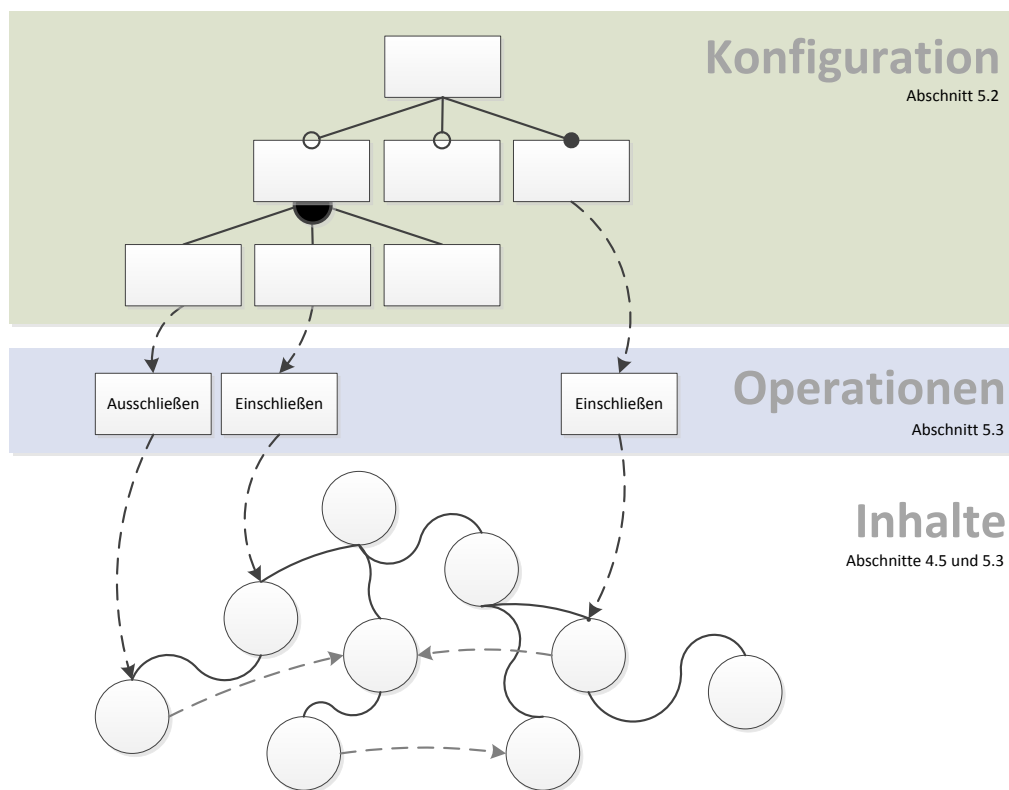


Abbildung 5.1: Überblick über Feature-basiertes Tailoring

Auf die einzelnen Bestandteile gehen wir in diesem Kapitel im Detail ein. Ein Überblick über dieses Kapitel ist in Abbildung 5.2 dargestellt:

Im ersten Teil dieses Kapitels entwickeln wir das Konfigurationsmodell zur Charakterisierung eines Projekts.

Im zweiten Abschnitt dieses Kapitels entwickeln wir – unter Rückgriff auf das einfache Meta-Metamodell für Vorgehensmodelle – eine Semantik für Tailoring-Operationen.

Im dritten Teil dieses Kapitels führen wir das Modell zur Charakterisierung des Projekts zusammen mit den Tailoring-Operationen und den durch das einfache Meta-Metamodell beschriebenen Inhalten eines Vorgehensmodells.

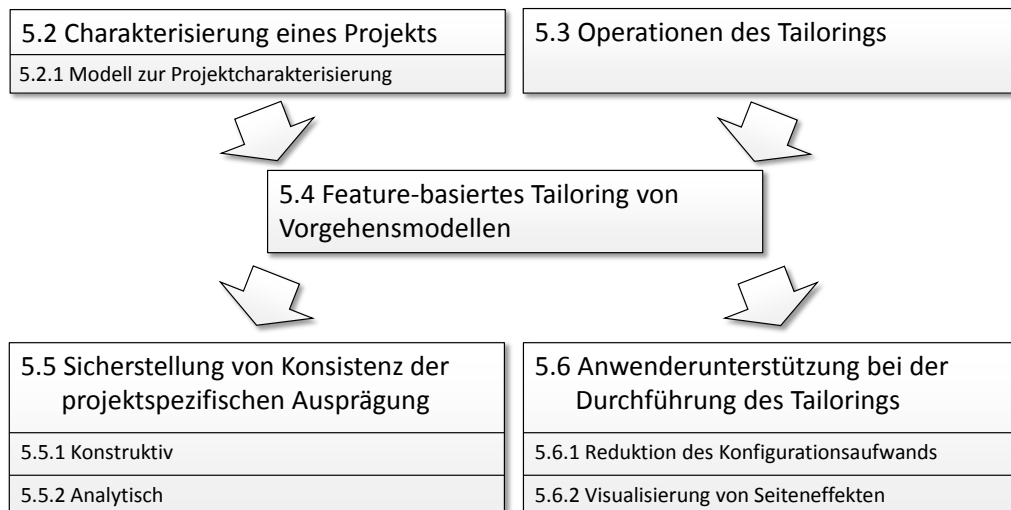


Abbildung 5.2: Struktur des Kapitels 5

Im vierten Teil des Kapitels befassen wir uns mit Verfahren, die zur Sicherstellung einer konsistenten Struktur und der inhaltlichen Sinnhaftigkeit einer projektspezifischen Ausprägung des Vorgehensmodells angewendet werden können. Da Bedingungen an die strukturelle Integrität und inhaltliche Sinnhaftigkeit stark vom jeweiligen konkreten Vorgehensmodell abhängen, stellen wir einige allgemeine Verfahren dar, die auf die Erfordernisse eines konkreten Vorgehensmodells übertragen werden können.

Teil fünf dieses Kapitels stellt zwei Verfahren zur Unterstützung des Anwenders bei der Durchführung des Tailorings basierend auf dem hier entwickelten Ansatz dar.

5.2 Charakterisierung eines Projekts

In Abschnitt 2.5 haben wir existierende Verfahren zur projektspezifischen Anpassung von Vorgehensmodellen diskutiert. Allen Verfahren ist gemein, dass die Situation des Projekts bei der Anpassung eine große Rolle spielt.

Fragen zur Charakterisierung

Im Folgenden gehen wir davon aus, dass die Beschreibung der Situation eines Projekts auf Basis einer festen Menge von Charakteristika möglich ist (vergleiche hierzu auch Abschnitt 2.4 und den Kriterienkatalog in Anhang A). Jedes Kriterium stellt dabei eine geschlossene Frage dar, d.h. die Antwortmöglichkeiten sind vorgegeben [Por09]. Für die Antwortmöglichkeiten sind dabei die in Abbildung 5.3 jeweils mit beispielhaften Werten dargestellten Skalen vorstellbar:

Ja-/Nein-Fragen Beispiele hierfür sind: Hat das zu entwickelnde System eine Benutzerschnittstelle? Ist im Projekt ein Altsystem zu berücksichtigen? Soll im Projekt kaufmännisches Projektmanagement durchgeführt werden?

Fragen mit kontinuierlichem, unbeschränktem Wertebereich Beispiele für diese Art von Fragen sind: Was ist die Projektlaufzeit? Wie groß ist das Projektbudget? Wie viele Mitarbeiter sind am Projekt beteiligt? Diese Fragen haben zwar streng genommen einen diskreten Wertebereich. Wir betrachten

5.2 Charakterisierung eines Projekts

aber etwa die Anzahl Mitarbeiter als quasi kontinuierlich, da fraglich ist, welchen Unterschied die Abstufung zwischen 52 oder 53 Mitarbeitern für die Charakterisierung des Projekts machen würde.

Einschätzungsfragen mit kontinuierlichem, beschränktem Wertebereich Beispiele für diese Fragen sind: Wie gut ist das Team mit der Domäne vertraut? Wie klar und eindeutig ist der Projektvorschlag? Wie hoch ist die Fluktuation im Projektteam? Welche Ansprüche bestehen an Safety und Security des zu entwickelnden Systems?

Einschätzungsfragen mit diskretem, beschränktem Wertebereich Fragen mit einem derartigen Wertebereich entsprechen den Einschätzungsfragen mit kontinuierlichem Wertebereich mit dem Unterschied, dass der Wertebereich durch eine fixe Zahl von Antwortmöglichkeiten „diskretisiert“ wird.

Auswahlfragen mit festem Wertebereich Beispiele für diese Art von Fragen sind: Welche Programmiersprache wird im Projekt verwendet? Was ist der Projekt-/Servicetyp? Was ist die Art der technischen Lösung?

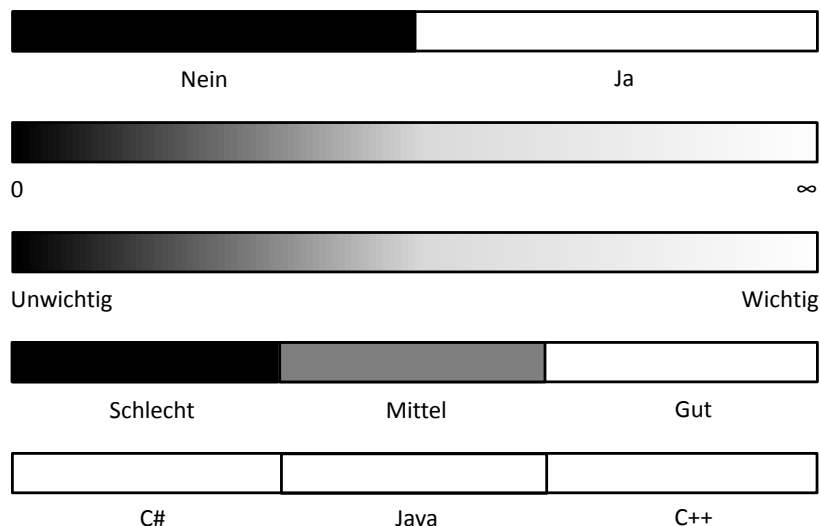


Abbildung 5.3: Beispiele für Antwortmöglichkeiten

Die Situationsbeschreibung soll zur projektspezifischen Konfiguration des Vorgehensmodells durch Auswahl geeigneter Inhalte dienen. Für diesen Zweck ist eine Charakterisierung basierend auf (quasi) kontinuierlichen Skalen nicht gut geeignet. Würde es einen Unterschied für die projektspezifische Ausprägung des Vorgehensmodells machen, ob die Anzahl der Mitarbeiter 52 oder 53 ist? Was wäre die Wirkung auf das Vorgehensmodell, wenn die Klarheit des Projektvorschlags auf einer Skala von 1 (nicht klar) bis 100 (klar) bei 67 eingeschätzt würde? Aus diesem Grund schränken wir die Antwortmöglichkeiten ein auf Ja-/Nein-Fragen und Fragen mit der Möglichkeit zur Auswahl aus einer festgelegten Menge von Antworten.

Fragen mit einem (quasi) kontinuierlichen Wertebereich bilden wir auf Fragen mit einer festgelegten Anzahl von Antworten ab.

Für die Anzahl Mitarbeiter im Projekt könnten derartige Antwortmöglichkeiten je nach Maximalgröße eines Teams in der Organisation beispielsweise so aussehen: 0-10, 10-50, 50-100. Alternativ könnte die Frage umformuliert werden zu *Wie groß ist das Team?* mit den Antwortmöglichkeiten *klein, mittel, groß*.

Für die Einschätzung der Klarheit des Projektvorschlags eignen sich z. B. folgende Antwortmöglichkeiten: *nicht klar*, *klar mit einigen offenen Punkten*, *völlig klar*. Sogar für diese beschränkte Anzahl ist fraglich, welche (abgestufte) Auswirkung auf die Inhalte des Vorgehensmodells die jeweiligen Möglichkeiten zur Einschätzung der Projektsituation haben sollten.

Abhängigkeit zwischen Charakteristika

Viele der vorstellbaren Kriterien zur Situationsbeschreibung beeinflussen sich gegenseitig oder hängen voneinander ab. Im Kriterienkatalog in Anhang A sind denkbare Querbezüge zu vielen der Kriterien angegeben. Die Querbezüge lassen sich in folgende Klassen einteilen:

Eine Antwort bedingt eine andere Antwort. Lautet der Projekt-/Servicetyp z. B. *Migration*, ist davon auszugehen, dass ein Altsystem im Projekt zu berücksichtigen ist. Handelt es sich bei der Art der technischen Lösung um eine *Webanwendung*, so hat diese eine *Benutzerschnittstelle*:

$$\text{Servicetyp} \mid_{\text{Migration}} \Rightarrow \text{Altsystem} \mid_{\text{Ja}}$$

$$\text{Technische Lösung} \mid_{\text{Webanwendung}} \Rightarrow \text{Benutzerschnittstelle} \mid_{\text{Ja}}$$

Eine Antwort verhindert eine andere Antwort. Lautet der Projekt-/Servicetyp z. B. *Beratung*, wird überhaupt keine technische Lösung erstellt und alle diesbezüglichen Fragen können und sollten nicht mehr beantwortet werden. Weiterhin können bedingende Abhängigkeiten zwischen Ja-/Nein-Fragen auch negiert formuliert werden. Beispielsweise könnte es eine Forderung sein, dass ein knappes Projektbudget kaufmännisches Projektmanagement erforderlich macht. Damit würde das knappe Projektbudget die Antwortmöglichkeit *Nein* zur Frage nach kaufmännischem Projektmanagement ausschließen.

$$\text{Servicetyp} \mid_{\text{Beratung}} \Rightarrow \neg \text{Technische Lösung}$$

$$\text{Servicetyp} \mid_{\text{Beratung}} \Rightarrow \neg \text{Benutzerschnittstelle} \mid_{\text{Ja}}$$

$$\text{Projektbudget} \mid_{\text{knapp}} \Rightarrow \neg \text{Kaufmännisches Projektmanagement} \mid_{\text{Nein}}$$

Eine Antwort bedingt implizit eine andere Antwort. Handelt es sich z. B. um ein Projekt mit dem Projekt-/Servicetyp *Weiterentwicklung* eines Altsystems, kann davon ausgegangen werden, dass die fachliche und technische Art des Altsystems auch bereits feststehen. Allerdings können die Antwortmöglichkeiten für die fachliche und technische Art des Systems nicht eingeschränkt werden. Durch den Projekt-/Servicetyp z. B. *Weiterentwicklung* steht zwar fest, dass nur eine Antwortmöglichkeit für die Art der fachlichen und technischen Lösung in Frage kommt, es kann aber nicht abgeleitet werden, welche das ist.

Mit der Betrachtung der möglichen Fragen zur Charakterisierung eines Projekts und möglichen Abhängigkeiten zwischen diesen haben wir die Grundlage gelegt für ein *Modell* zur Beschreibung der Projektsituation.

5.2.1 Modell zur Projektcharakterisierung

Um das projektspezifische Tailoring basierend auf der Charakterisierung durch Fragen (siehe Abschnitt 5.2) durchführen zu können, formalisieren wir die Vereinbarungen zu möglichen Antworten und zu möglichen Abhängigkeiten in einem Modell. Wir orientieren uns hierbei an der Feature-Modellierung (zur Rekapitulation siehe Abschnitt 2.6). Feature-Modelle sind gut geeignet zur Darstellung der Abhängigkeiten zwischen Charakteristika und erlauben durch ihre mathematische Fundierung die Prüfung der Gültigkeit einer gegebenen Antwortmenge (*Konfiguration*).

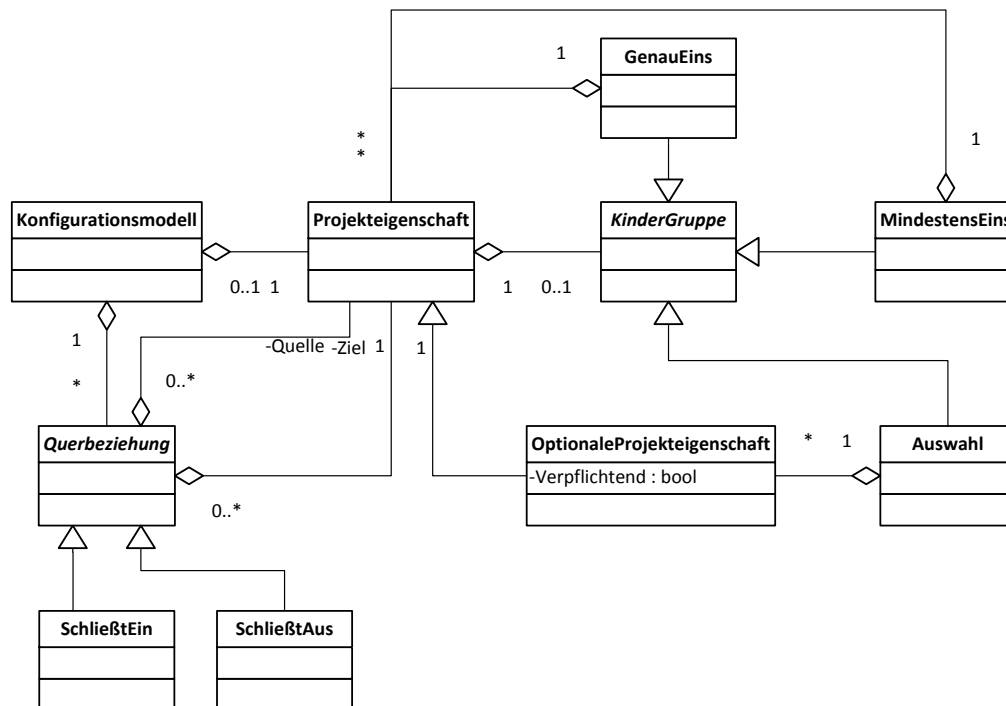


Abbildung 5.4: Metamodell für das Konfigurationsmodell zum Tailoring

In den folgenden Ausführungen verwenden wir das in Abbildung 5.4 dargestellte Metamodell des Konfigurationsmodells zum Tailoring. Das Metamodell beschreibt einen Featurebaum entsprechend der Ausführungen in Abschnitt 2.6. Die Klassen des Metamodells im Einzelnen sind:

Konfigurationsmodell Das Konfigurationsmodell repräsentiert das gesamte Modell der Charakterisierung. Es besteht aus einem Baum von Projekteigenschaften (Features) und einer Menge von Querbeziehungen zwischen Projekteigenschaften (diese entsprechen den *Cross-Tree Constraints* aus Abschnitt 2.6).

Projekteigenschaft Eine Projekteigenschaft repräsentiert entweder eine Frage zur Charakterisierung, eine Antwortmöglichkeit zu einem Charakteristikum oder wird zur hierarchischen Gruppierung der Charakteristika verwendet.

KinderGruppe Diese abstrakte Basisklasse dient zur Darstellung der Baumstruktur der Charakteristika. Jede Projekteigenschaft (Feature) kann Kind-Eigenschaften (Kind-Features) beinhalten, die über eine der Spezialisierungen von KinderGruppe gruppiert sind.

- GenauEins** Die GenauEins Gruppierung gibt an, dass die Kind-Eigenschaften in einer XOR-Beziehung zueinander stehen. D.h. aus der Menge der Kind-Eigenschaften kann nur genau eine ausgewählt werden.
- MindestensEins** Eine MindestensEins Gruppierung bedeutet, dass mindestens eine der Kind-Eigenschaften gewählt sein muss. Die Kind-Eigenschaften stehen also in einer OR-Beziehung zueinander.
- Auswahl** Eine Auswahl-Gruppe kann Kind-Eigenschaften vom Typ Optionale-Projekteigenschaft beinhalten. OptionaleProjekteigenschaften, die als Verpflichtend markiert sind, sind dabei immer in der Gruppe enthalten. OptionaleProjekteigenschaften, die nicht als Verpflichtend markiert sind, können gewählt werden oder nicht.
- OptionaleProjekteigenschaft** Diese Spezialisierung der Klasse Projekteigenschaft ist in der Gruppierung Auswahl enthalten und trägt gegenüber der Projekteigenschaft ein Attribut zur Angabe, ob die entsprechende Eigenschaft immer Teil der übergeordneten Gruppe ist (*Verpflichtend = wahr*) oder ob es optional Teil der Gruppe sein kann (*Verpflichtend = falsch*).
- Querbeziehung** Diese abstrakte Basisklasse repräsentiert eine Querbeziehung zwischen Projekteigenschaften oder Antwortmöglichkeiten (Features). Sie verweist auf eine Quell-Eigenschaft (linke Seite der Abhängigkeit) und eine Ziel-Eigenschaft (rechte Seite der Abhängigkeit). Im Gegensatz zu manchen Feature-Modellen [CE00, KCH⁺90] erlauben wir in unserem Modell der Einfachheit halber keine Querbeziehungen mit mehr als einer Quell- oder Ziel-Eigenschaft. Für einen derartigen Fall müsste es in unserem Modell mehrere Instanzen von Querbeziehung für jede Paarung geben. Diese Vereinfachung kann in Kauf genommen werden, denn die Erfahrung bei der praktischen Anwendung des feature-basierten Tailorings – insbesondere in unseren Fallstudien in Kapitel 7 – hat gezeigt, dass Querbeziehungen in der Realität selten eingesetzt werden (müssen).
- SchließtEin** Diese Spezialisierung von Querbeziehung zeigt an, dass die Querbeziehung zwischen den teilnehmenden Features eine bedingende ist, d.h. aus der Wahl der Quell-Eigenschaft folgt die Wahl der Ziel-Eigenschaft.
- SchließtAus** Diese Spezialisierung der Klasse Querbeziehung repräsentiert eine ausschließende Querbeziehung, d.h. die Wahl der Quell-Eigenschaft verhindert die Wahl der Ziel-Eigenschaft.

Abbildung 5.5 veranschaulicht den Zusammenhang zwischen dem Metamodell des Konfigurationsmodells und der üblicherweise verwendeten Notation für Featurebäume. Der Übersichtlichkeit halber sind nicht für alle Projekteigenschaften (sprich: Features) die Instanz-Beziehungen eingezeichnet.

Im Folgenden skizzieren wir, wie die Charakterisierung eines Projekts im Sinne des Abschnitts 5.2 mit Hilfe dieses Modells darstellbar ist.

Abbildung der Projektcharakteristika auf das Konfigurationsmodell

Jede Projekteigenschaft im Konfigurationsmodell repräsentiert für sich genommen eine Boolesche Variable. Diese Tatsache dient als Grundlage für die Abbildung der Fragen zur Charakterisierung auf das Konfigurationsmodell. Zur Rekapitulation: wir gehen davon aus, dass die Charakterisierung durch Beantwortung von Ja-/Nein-Fragen, Einschätzungsfragen mit einem festen Wertebereich oder Auswahlfragen mit einem festen Wertebereich erfolgt.

5.2 Charakterisierung eines Projekts

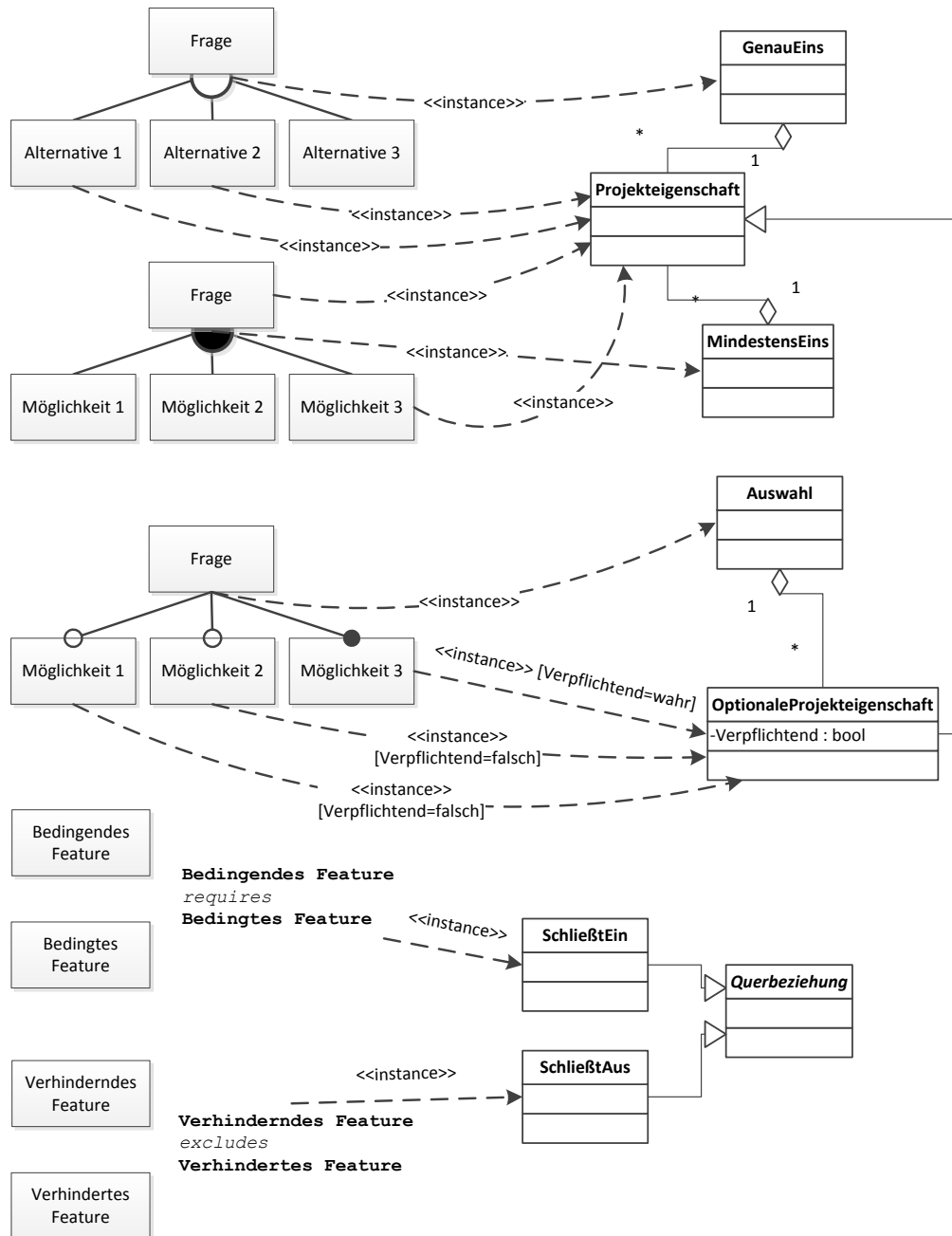


Abbildung 5.5: Zusammenhang zwischen Metamodell und Notation für Feature-Modelle

Ja-/Nein-Frage: Ein durch eine Ja-/Nein-Frage zu beantwortendes Projektcharakteristikum kann direkt auf eine Eigenschaft im Konfigurationsmodell (ein Feature) abgebildet werden. Wird die Frage mit Ja beantwortet, ist die entsprechende Eigenschaft in der Konfiguration gewählt, andernfalls nicht. Für Projektcharakteristika mit zwei Antwortmöglichkeiten, bei denen die Bedeutung der Wahl oder Nicht-Wahl nicht offensichtlich ist, kann die entsprechende Frage entweder geeignet umformuliert werden oder es werden zwei Kind-Eigenschaften in einer XOR-Verknüpfung eingeführt. Man stelle sich das Projektcharakteristikum *Teamgröße* mit den möglichen Werten *klein* und *groß* vor. Entweder werden zwei XOR-verknüpfte Eigenschaften *Klein* und *Groß* als Kinder von *Projektgröße* eingeführt oder das Charakteristikum wird umformuliert, beispielsweise *Handelt es sich um ein großes Team?*

Einschätzungsfrage mit festem Wertebereich: Bei einer Einschätzungsfrage mit festem Wertebereich kann immer nur genau eine der Antwortmöglichkeiten gewählt werden. Fragen dieses Typs werden auf eine die Frage repräsentierende Projekteigenschaft mit XOR-verknüpften Kindern für jede der Antwortmöglichkeiten abgebildet. Im Hinblick auf das Metamodell aus Abbildung 5.4 bedeutet diese Abbildung die Gruppierung der Kind-Features in einer *GenauEins* Gruppe.

Auswahlfrage mit festem Wertebereich: Auswahlfragen mit festem Wertebereich werden auf eine Eigenschaft für die Frage und Kind-Eigenschaften für die Antwortmöglichkeiten abgebildet. Abhängig von der Art des Charakteristikums sind die Kinder durch XOR, OR oder Optional/Verpflichtend verknüpft.

Abbildung von Abhängigkeiten zwischen Charakteristika: Für die Abbildung einer bedingenden Abhängigkeit zwischen zwei Charakteristika gibt es folgende Möglichkeiten:

- Eine Eigenschaft im Konfigurationsmodell ist nur dann wählbar, wenn ihre Eltern-Eigenschaft gewählt ist – d.h. zwischen Kind- und Eltern-Eigenschaft herrscht eine bedingende Abhängigkeit. Dieser Umstand kann zur Darstellung der bedingenden Abhängigkeit zwischen Projektcharakteristika genutzt werden. Man kann sich beispielsweise vorstellen, dass die Wahl einer Programmiersprache nur dann erfolgen soll, wenn das Projekt einen Entwicklungsanteil hat. Die Wahl einer Programmiersprache wird daher sinnvollerweise im Teilbaum unterhalb des entsprechenden Projekt-/Servicetyp angesiedelt.
- Durch verpflichtende Kind-Eigenschaften kann eine bedingende Abhängigkeit zwischen Eltern- und Kind-Eigenschaft ausgedrückt werden (im Metamodell Konfigurationsmodell aus Abbildung 5.4 entspricht dies einer als *Verpflichtend* markierten *OptionaleProjekteigenschaft* in einer *Auswahl*-Gruppe). Die bedingende Abhängigkeit zwischen Eltern- und Kind-Eigenschaft kann zur Darstellung der bedingenden Abhängigkeit zwischen zwei Projektcharakteristika genutzt werden.
- Kann die bedingende Beziehung nicht sinnvoll über die Baumstruktur ausgedrückt werden, wird zu ihrer Darstellung eine bedingende Querbeziehung zwischen bedingendem und bedingtem Projektcharakteristikum eingeführt.

Für die Darstellung einer verhindernden Abhängigkeit zwischen zwei Projektcharakteristika im Konfigurationsmodell gibt es folgende Möglichkeiten:

- Sind die Projekteigenschaften im Baum in einer XOR-Gruppierung gruppiert, kann nur genau eines der Kinder in der Gruppe gewählt werden.

5.3 Operationen des Tailorings

Diese Eigenschaft von Featurebäumen kann zur Darstellung des gegenseitigen Ausschlusses von Projektcharakteristika verwendet werden.

- Lässt sich der gegenseitige Ausschluss nicht sinnvoll über die Baumstruktur ausdrücken, kann eine Querbeziehung zwischen den betreffenden Projekteigenschaften eingeführt werden.

Insgesamt ergeben sich für die Abbildung der Projektcharakteristika auf das Konfigurationsmodell die in den Tabellen 5.1 und 5.2 veranschaulichten Alternativen.

Die durch das Konfigurationsmodell beschriebene Charakterisierung eines Projekts stellt die Grundlage für die projektspezifische Anpassung eines Vorgehensmodells dar. Im folgenden Abschnitt befassen wir uns mit der Wirkung der Konfiguration auf die Inhalte des Vorgehensmodells.

5.3 Operationen des Tailorings

Im folgenden Abschnitt entwickeln wir die Operationen, die durch Konfiguration auf den Vorgehensmodellinhalten durchgeführt werden können. Die Operationen stellen die Verbindung her zwischen dem Konfigurationsmodell aus Abschnitt 5.2 und den Vorgehensmodellinhalten.

Wie bereits in Abschnitt 2.4 dargelegt, kommen laut [XR08] für das Tailoring eines Vorgehensmodells prinzipiell folgende Operationen in Frage:

Add Ein Inhalt ist im Vorgehensmodell noch nicht vorhanden und sollte hinzugefügt werden.

Downsize Ein Inhalt ist zu umfangreich für die Situation des Projekts. Der Umfang (beispielsweise der Beschreibung) sollte reduziert werden.

Drop/Skip Der Inhalt wird in der Projektsituation nicht benötigt und sollte aus dem Vorgehensmodell entfernt werden.

Expand/Concretize Inhalte sind für die Erfordernisse der Projektsituation nur unzureichend beschrieben. Sie sollten detailliert werden.

Redefine Ein Inhalt ist im Vorgehensmodell vorhanden, hat aber in der Projektsituation eine andere Bedeutung. Es muss für die Erfordernisse der Projektsituation umdefiniert werden.

Replace Vorhandene Inhalte im Vorgehensmodell müssen für die Projektsituation durch andere Inhalte ersetzt werden.

Auch wenn man durch die Namen der Operationen und deren informelle Beschreibung eine Vorstellung von ihrer Bedeutung hat, bleibt unklar, wie sich diese Operationen in einem Vorgehensmodell konkret auswirken.

Um die Tailoring-Operationen mit einer Semantik zu versehen, rekapitulieren wir zunächst das einfache Meta-Metamodell für Vorgehensmodelle aus Kapitel 4.

Rekapitulation des einfachen Meta-Metamodells für Vorgehensmodelle: Das vollständige Meta-Metamodell für Vorgehensmodelle ist in Abbildung 5.6 dargestellt.

Wie in Kapitel 4 ausgeführt, ist eine Vereinbarung des Modells, dass Tailoring-Operationen nur auf Vorgehensmodell-Elemente vom Typ *VariablesElement* anwendbar sind.


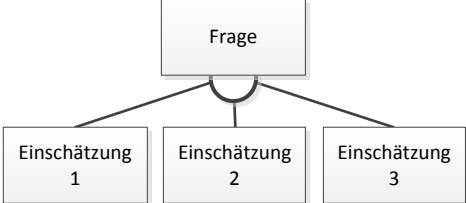
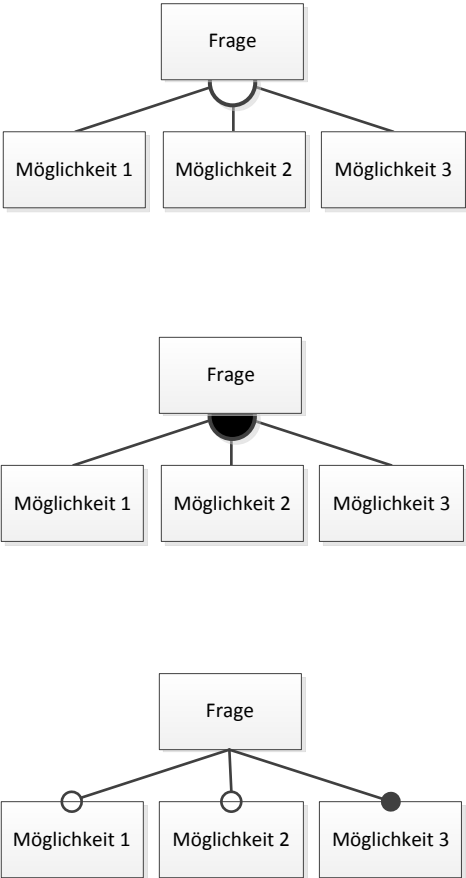
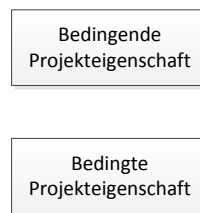
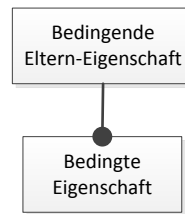
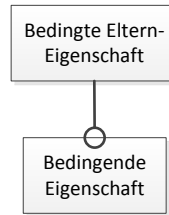
Art der Frage	Abbildung im Konfigurationsmodell
Ja-/Nein	
Einschätzung	
Auswahl	

Tabelle 5.1: Abbildung von Projektcharakteristika auf Konfigurationsmodell

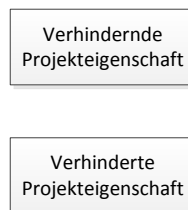
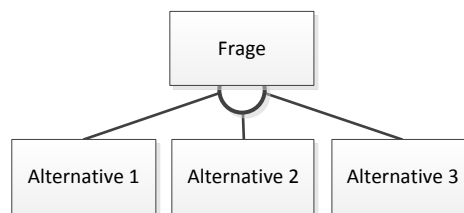
Art der Abhängigkeit Abbildung im Konfigurationsmodell

Bedingend



Bedingende Projekteigenschaft
requires
Bedingte Projekteigenschaft

Verhindernd



Verhindernde
Projekteigenschaft
excludes
Verhinderte
Projekteigenschaft

Tabelle 5.2: Abbildung von Querbeziehungen auf Konfigurationsmodell

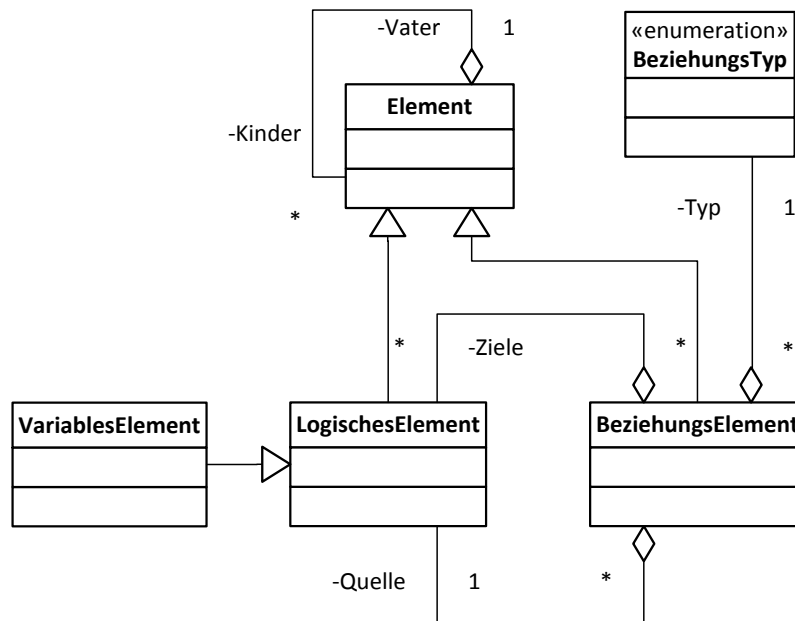


Abbildung 5.6: Einfaches Meta-Metamodell für Vorgehensmodelle

Zur Beschreibung der Semantik der Tailoring-Operationen verwenden wir die in Abschnitt 4.5 vereinbarten Mengen der Vorgehensmodellinhalte (zur Rekapitulation: Die Menge *Vorgehensmodell* enthält alle Elemente; die Menge *VariableElemente* enthält alle Elemente vom Typ *VariablesElement*). Wir definieren zunächst die folgende einfache Abbildung:

$$\text{Include} : \{\text{wahr}, \text{falsch}\} \times \text{VariableElemente} \rightarrow \text{Vorgehensmodell}$$

Entsprechend der in Abschnitt 4.5 vereinbarten Semantik spezifizieren wir die Abbildung wie folgt:

$$\text{Include} : (\text{wahr}, ve) \mapsto \text{Vorgehensmodell}$$

$$\text{Include} : (\text{falsch}, ve) \mapsto \text{Vorgehensmodell} \setminus \text{Tailoring}(ve)$$

Mit $VE := \text{VariableElemente}$ und $VM := \text{Vorgehensmodell}$ vereinbaren wir die Semantik der Tailoring-Operationen nun wie folgt:

5.4 Feature-basiertes Tailoring von Vorgehensmodellen

$$Add : VE \rightarrow VM, v \mapsto Include(wahr, v)$$

$$Downsize : VE \rightarrow VM, v \mapsto Include(falsch, u \in Kinder(v)) \vee \\ Replace(v, u \in VE \mid u \neq v)$$

$$Drop : VE \rightarrow VM, v \mapsto Include(falsch, v)$$

$$Expand : VE \rightarrow VM, v \mapsto Include(wahr, u \in Kinder(v)) \vee \\ Replace(v, u \in VE \mid u \neq v)$$

$$Redefine : Replace(v, u \in VE \mid u \neq v)$$

$$Replace : VE \times VE \rightarrow VM, (v, u) \mapsto Include(falsch, v) \wedge Include(wahr, u)$$

Die Semantik von *Add*, *Drop* und *Replace* sollte selbsterklärend sein. *Redefine* definieren wir als Ersetzung des umzudefinierenden, variablen Elements durch ein anderes. *Downsize* wird definiert als entweder die Entfernung eines Teilbaums unterhalb des zu reduzierenden Elements oder als Ersetzung durch ein entsprechend reduziertes, variables Element. Die Operation *Expand* ist analog definiert. Mit der so definierten Semantik der Tailoring-Operationen können diese alle zurückgeführt werden auf die Operation:

$$Include(b \in \{wahr, falsch\}, v \in VariableElemente)$$

Damit stellt diese Operation die Verbindung dar zwischen dem in Abschnitt 5.2 definierten Konfigurationsmodell und den Inhalten des Vorgehensmodells. Im folgenden Abschnitt setzen wir die Bausteine zusammen.

5.4 Feature-basiertes Tailoring von Vorgehensmodellen

In diesem Abschnitt fügen wir das Konfigurationsmodell aus Abschnitt 5.2, die Tailoring-Operationen aus Abschnitt 5.3 und die Vorgehensmodellinhalte zu einem integrierten Modell des feature-basierten Tailorings zusammen. Die Vorgehensmodellinhalte sind in jedem Vorgehensmodell unterschiedlich beschrieben und strukturiert. Wir verwenden zur Beschreibung derselben deshalb das einfache Meta-Metamodell aus Abschnitt 4.5 (für eine Übersicht über das ganze Meta-Metamodell siehe auch Abbildung 5.6).

Zur Darstellung der Operationen definieren wir zwei UML Stereotypen [OMG07] für Abhängigkeiten zwischen Projekteigenschaften im Konfigurationsmodell und variablen Elementen: *Einschließen* und *Ausschließen*.

Zur Definition der Stereotypen gehen wir davon aus, dass durch eine Abbildung *IstGewählt* : *Projekteigenschaft* \mapsto $\{wahr, falsch\}$ auf einer *Projekteigenschaft* abgefragt werden kann, ob diese in der Konfiguration gewählt ist oder nicht.

Eine mit diesen Stereotypen ausgezeichnete Abhängigkeit soll folgende Bedeutung haben:

Einschliessen : $\text{Projekteigenschaften} \times \text{VariableElemente} \rightarrow \text{Vorgehensmodell}$,
 $(e, v) \mapsto \text{Include}(e.\text{IstGewählt}, v)$

Ausschliessen : $\text{Projekteigenschaften} \times \text{VariableElemente} \rightarrow \text{Vorgehensmodell}$,
 $(e, v) \mapsto \text{Include}(\neg e.\text{IstGewählt}, v)$

Mit diesen Stereotypen ergibt sich das in Abbildung 5.7 dargestellte Gesamtbild des feature-basierten Tailorings von Vorgehensmodellen.

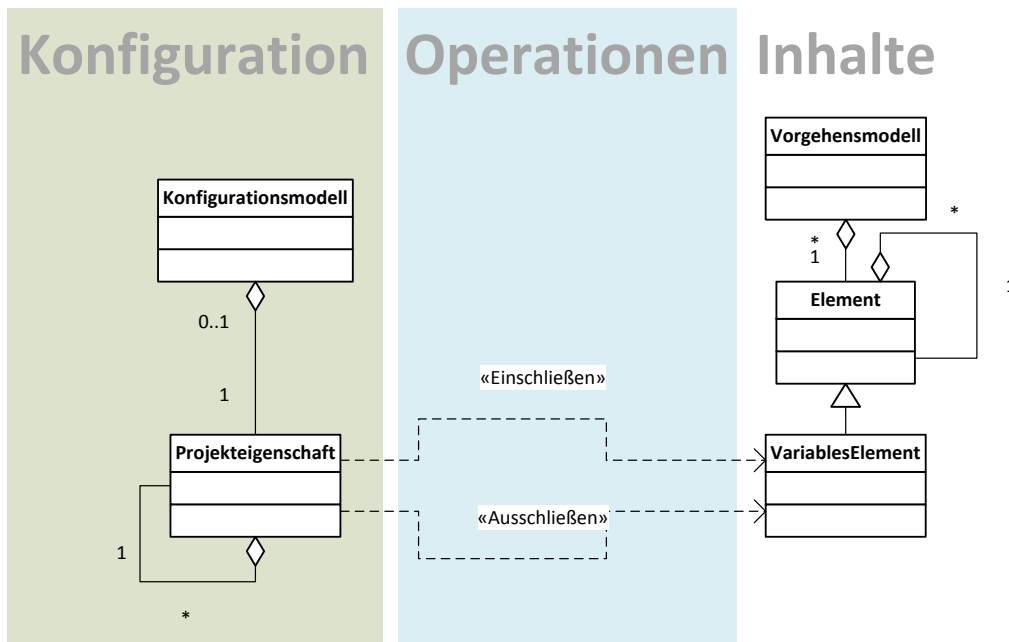


Abbildung 5.7: Vereinfachtes Metamodell Feature-basiertes Tailoring

Aus Gründen der Übersichtlichkeit, und weil wir für die allgemeinen Betrachtungen von konkreten Vorgehensmodell-Klassen abstrahieren, sind die Meta-Ebenen in Abbildung 5.7 vermischt: Die Abbildung stellt die Meta-Modell-Ebene (M2 – vergleiche Abschnitt 2.2.1) eines Vorgehensmodells dar. Die Klassen *Element* und *VariablesElement* sind als Instanzen der gleichnamigen Meta-Metamodell-Klassen des Meta-Metamodells (M3) aus Abbildung 5.6 zu verstehen. Weiterhin sind in der Abbildung nur die wesentlich am Tailoring beteiligten Elemente des Vorgehensmodells dargestellt (ein Überblick über das vollständige Modell findet sich in Abbildung 5.8 – auch in dieser Abbildung befindet sich der untere, graue Bereich auf Meta-Modell-Ebene M2, während der obere, weiße Bereich auf Ebene M3 angesiedelt ist).

Das Vorgehensmodell besteht aus hierarchisch organisierten Inhalten, namentlich den Elementen (in Abbildung 5.7 auf der rechten Seite dargestellt). Beziehungen sind in der Abbildung der Übersichtlichkeit halber nicht eingezeichnet. Auf der anderen Seite steht das Konfigurationsmodell mit seinem Baum von Projekteigenschaften. Die Eigenschaften im Konfigurationsmodell und die Inhaltselemente im Vorgehensmodell können über einschließende oder ausschließende Beziehungen miteinander verbunden sein. Nimmt ein Inhaltselement nicht an einer derartigen Konfigurationsbeziehung teil, ist es automatisch in allen projektspezifischen Ausprägungen des Vorgehensmodells enthalten. Im

5.4 Feature-basiertes Tailoring von Vorgehensmodellen

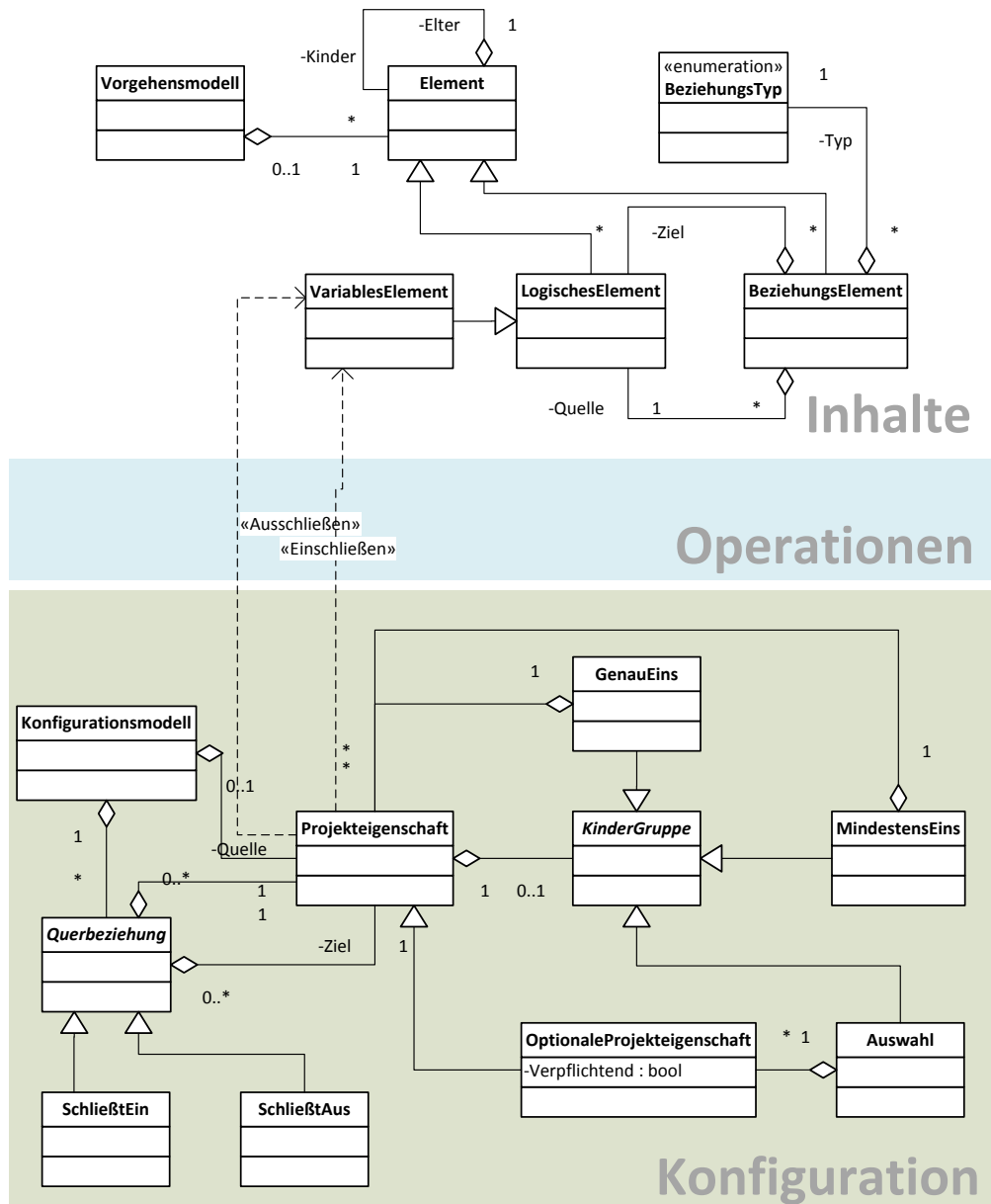


Abbildung 5.8: Komplettes Metamodell Feature-basiertes Tailoring

Sinne der Produktlinienmodellierung entsprechen Inhalte, die an keiner ein- oder ausschließenden Konfigurationsbeziehung teilnehmen, den *Commonalities* [BKPS04]. Inhaltselemente, die potenziell von der projektspezifischen Konfiguration betroffen sein können, entsprechen in dieser Terminologie den (*controlled*) *Variabilities*.

In Abbildung 5.7 und in Abbildung 5.8 sind das Vorgehensmodell und das Konfigurationsmodell einzig über die Operationen verbunden. In allen Umsetzungen des feature-basierten Tailoring (siehe Kapitel 7) sind beide Modelle in einem physischen Modell integriert. Im SE Book (vergleiche Abschnitt 7.2) ist das Konfigurationsmodell Teil des XML-Baums dieses Modells. Auch in der beispielhaften Realisierung für das V-Modell XT (vergleiche Abschnitt 7.3) wurde das Konfigurationsmodell in die XML-Beschreibung dieses Vorgehensmodells aufgenommen. Das Konfigurationsmodell kann daher in beiden Realisierungen als ein *Submodell* des Vorgehensmodells (vergleiche Abschnitt 2.2.1) aufgefasst werden. Das Vorgehensmodell selbst trägt die Informationen zur eigenen Konfiguration.

„Richtung“ der Operationen

In den Abbildungen 5.7 und 5.8 ist die Richtung der Operations-Abhängigkeiten von der Projekteigenschaft zum ein- oder ausschließenden Vorgehensmodell-Element eingezeichnet. Dies entspricht der Realisierung im V-Modell XT. Hier referenzieren die *Projekttypen*, *Projekttypvarianten* und *Projektmerkmale* die zu konfigurierenden Modellelemente *Vorgehensbausteine*, *Ablaufbausteine* und *Ablaufbausteinspezifikationen* (vergleiche auch Abschnitt 3.2.1). Im SE Book ist die Richtung der Referenzierung umgekehrt (siehe z. B. Abbildung 7.6): hier verweisen Inhaltselemente auf die konfigurierenden Projekteigenschaften. Das Inhaltselement „gibt selbst an“, an welcher Konfiguration es teilnimmt. Welche Richtung der Referenzierung zu empfehlen ist, hängt von der Art und Struktur des Vorgehensmodells ab. Im V-Modell XT ist die Anzahl der variablen Metamodel-Elemente mit 3 sehr überschaubar. Im SE Book hingegen kann nahezu jedes Meta-Modellelement an der Konfiguration teilnehmen. Würde hier die Richtung des V-Modell XT gewählt, müsste eine Projekteigenschaft Referenzen auf fast alle Metamodel-Elemente aufnehmen können.

Operationen auf Beziehungselementen

Aus Abbildung 5.8 geht hervor, dass Beziehungselemente nicht direkt an der Konfiguration teilnehmen können. Dies ist aber keine Beschränkung der Allgemeinheit. Wie Beziehungen Gegenstand der projektspezifischen Anpassung sein können, lässt sich am Beispiel der Beziehungen zwischen Produkten im V-Modell XT illustrieren. Abbildung 5.9 rekapituliert die Realisierung von $n : m$ Beziehungen zwischen Produkten im V-Modell XT und den Zusammenhang zwischen Metamodel-Elementen und dem einfachen Meta-Metamodel für Vorgehensmodelle aus Abschnitt 4.5.

Die Beziehungselemente *ErzeugendeAbhängigkeitVonProdukt* und *ErzeugendeAbhängigkeitZuProdukt* können nicht direkt am Tailoring teilnehmen. Sie werden allerdings zusammengehalten durch das logische Element *ErzeugendeAbhängigkeit*. Dieses könnte als *VariablesElement* gekennzeichnet werden und damit an der projektspezifischen Konfiguration teilnehmen. Würde die *ErzeugendeAbhängigkeit* aus der projektspezifischen Ausprägung des Vorgehensmodells entfernt,

5.4 Feature-basiertes Tailoring von Vorgehensmodellen

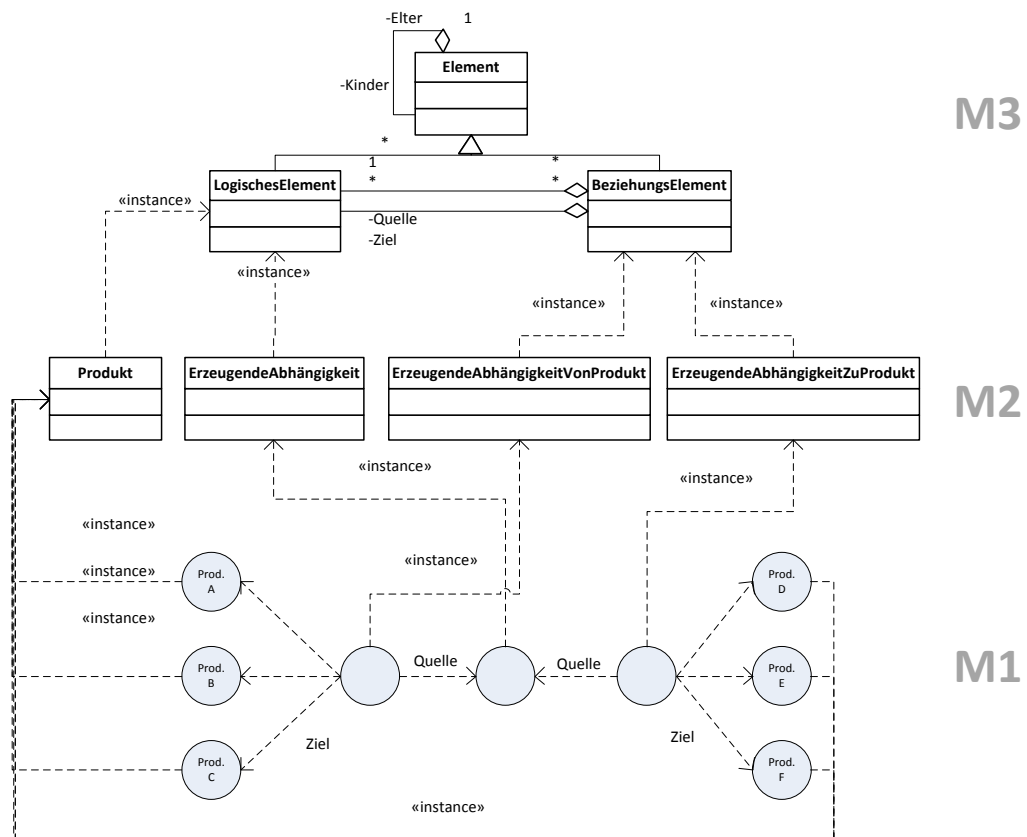


Abbildung 5.9: Rekapitulation: $n : m$ Beziehung zwischen Produkten

würden auch die beiden assoziierten Beziehungselemente entfernt, da die *Erzeugende Abhängigkeit* bei beiden Beziehungselementen jeweils das Quell-Element ist (vergleiche Abschnitt 4.5).

5.5 Sicherstellung von Konsistenz der projektspezifischen Ausprägung

Die Struktur und das Verhalten des feature-basierten Tailorings sind nun vollständig beschrieben. In diesem Abschnitt betrachten wir, wie Struktur und das Verhalten dazu genutzt werden können, eine strukturell und inhaltlich sinnvolle projektspezifische Ausprägung des Vorgehensmodells sicherzustellen.

Strukturelle Konsistenz

Unter struktureller Konsistenz verstehen wir insbesondere, dass die *Struktur* der projektspezifischen Ausprägung gültig im Sinne des Metamodells des Vorgehensmodells ist.

Zumindest ein struktureller *bad smell* ist es, wenn durch die Konfiguration „Waisen“ in der projektspezifischen Ausprägung entstehen können. Zur Illustration folgendes Beispiel: im V-Modell XT kann ein *Produkt* mehrere *Themen* beinhalten (vergleiche Abbildung 5.10). Angenommen, das *Produkt* wäre eine Instanz von *VariablesElement* – wäre also ein Variabilitätspunkt im Tailoring. Würde das Produkt aus der projektspezifischen Ausprägung entfernt, würde auch die entsprechende Beziehung *ThemaZuProdukt* entfernt (vergleiche Abschnitt 4.5 – das Produkt ist das einzige Ziel-Element der Beziehung). Damit sind Konfigurationen möglich, bei denen ein *Thema* keinem Produkt mehr zugeordnet ist. Das Thema wäre verwaist.

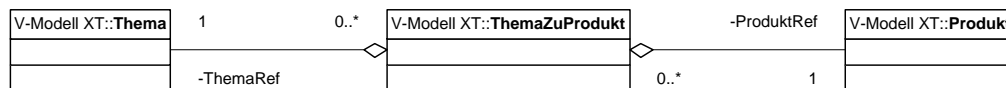


Abbildung 5.10: Beziehung zwischen Themen und Produkten im V-Modell XT

Ebenfalls als strukturelles Problem betrachten wir eine Konfiguration, die *Widersprüche* in der Struktur der Inhalte ergeben kann. Exemplarisch für einen strukturellen Widerspruch betrachten wir das Modell in Abbildung 5.11.

In der Baumstruktur der Vorgehensmodellinhalte ist das *Objekt 2* über die Komposition ein Teil von *Objekt 1* (es liegt im Teilbaum unterhalb von *Objekt 1*). Wird *Eigenschaft 1* bei der Konfiguration nicht gewählt, wird der Teilbaum unterhalb des *Objekt 1* aus der projektspezifischen Ausprägung entfernt. Insbesondere wird auch das *Objekt 2* entfernt. Wenn *Eigenschaft 2* in der Konfiguration aber gewählt wurde, entstünde ein Widerspruch: Sollte *Objekt 2* Teil der projektspezifischen Ausprägung sein, müssen auch die Elemente auf dem Pfad zur Wurzel des Baums eingeschlossen sein.

Inhaltliche Konsistenz

Unter inhaltlicher Konsistenz verstehen wir die Einhaltung inhaltlicher Regeln und Konventionen des Vorgehensmodells. Für das V-Modell XT gibt es etwa die

5.5 Sicherstellung von Konsistenz der projektspezifischen Ausprägung

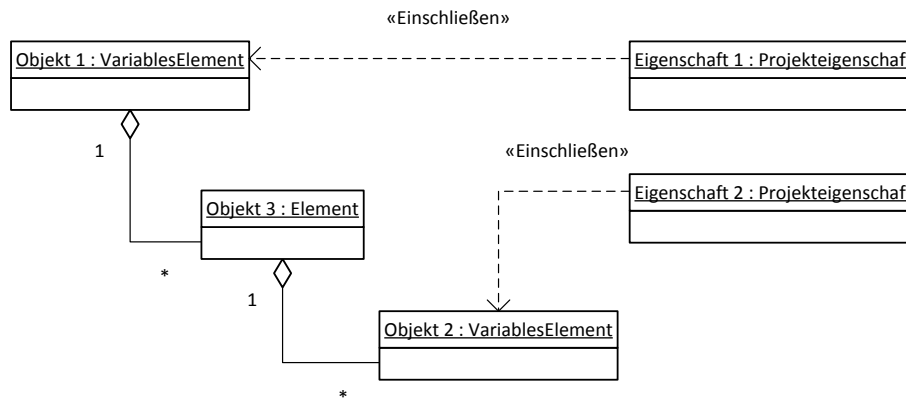


Abbildung 5.11: Struktureller Widerspruch in der Konfiguration

Konvention, dass zu jedem *Produkt* eine *Rolle* in einer verantwortenden Beziehung stehen muss. Derartige Regeln und Konventionen sind nicht im Metamodell ausgedrückt und müssen bei der Erstellung von Vorgehensmodellinhalten „manuell“ sichergestellt werden.

Für die obigen Beispiele skizzieren wir im Folgenden, wie die Struktur und das Verhalten des feature-basierten Tailorings genutzt werden können, um die Konsistenz sicherzustellen.

5.5.1 Konstruktive Sicherstellung

Das wichtigste Mittel der konstruktiven Sicherstellung von Konsistenz ist die Wahl geeigneter Variabilitätspunkte. Gilt¹ die Bedingung

$$\forall ve \in VariableElemente : Teilbaum(ve) \cap VariableElemente = \emptyset$$

so kann ein struktureller Konflikt in der Konfiguration, wie oben beschrieben und anhand von Abbildung 5.11 veranschaulicht, nicht auftreten. Die Forderung der Bedingung ist, dass variable Elemente sich nicht gegenseitig enthalten dürfen. Mit anderen Worten: die Inhalte der Variabilitätspunkte sind disjunkt.

Genau diese Art der konstruktiven Sicherstellung struktureller Konsistenz findet im V-Modell XT Anwendung: der zentrale Variabilitätspunkt ist der *Vorgehensbaustein*. In die Inhalte eines Vorgehensbausteins kann die Konfiguration (im Rahmen des Tailorings) nicht „hineingreifen“.

Das anhand der Beziehung zwischen *Thema* und *Produkt* in Abbildung 5.10 veranschaulichte Problem eines strukturellen *bad smells* kann durch rekursive Erweiterung der Algorithmen gelöst werden. Anstatt, wie in Abschnitt 4.5 beschrieben, nur aufgrund der Entfernung von Elementen verwaiste Beziehungselemente zu entfernen, müssten verwaiste Elemente des Vorgehensmodells rekursiv berechnet werden. Da dies nichts an den grundlegenden Mechanismen ändert, sparen wir uns an dieser Stelle die Skizzierung des entsprechenden Verfahrens. Zu beachten ist lediglich, dass durch die rekursive Berechnung zu entfernender Elemente die Grenzen der variablen Anteile im Vorgehensmodell ausgedehnt werden. Abbildung 5.12 veranschaulicht diese Ausdehnung durch

¹ unter Verwendung der Menge $Teilbaum(x)$ aus Abschnitt 4.5

Nachverfolgen verwaister Beziehungen (die Kreise seinen jeweils beliebige Inhaltselemente. Diese sind in einer hierarchischen Struktur angeordnet und sind zudem über Beziehungen miteinander verbunden).

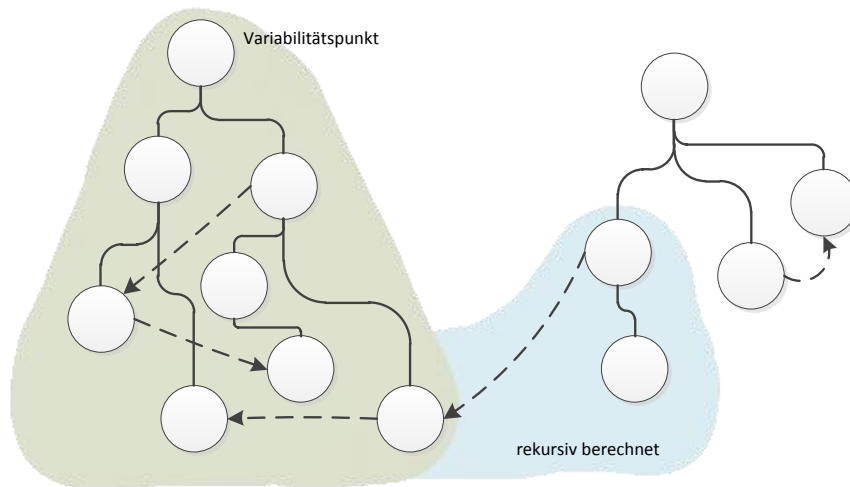


Abbildung 5.12: Ausdehnung variabler Modellanteile durch rekursive Nachverfolgung

Auch für die inhaltlichen Randbedingungen an das Tailoring (veranschaulicht am Beispiel „Jedes Produkt hat eine Rolle, die es verantwortet“) skizzieren wir Möglichkeiten der konstruktiven Sicherstellung:

- Ein erster Schritt zur Sicherstellung der Bedingung liegt in der Struktur des V-Modell XT: Rollen liegen hier *außerhalb* der Vorgehensbausteine und können damit insbesondere nicht aus der projektspezifischen Ausprägung des Vorgehensmodells entfernt werden (also $Rollen \cap (VariableElemente \cup Kinder(VariableElemente)) = \emptyset$). Damit wird vermieden, dass ein Produkt im Sinne der Bedingung verweisen kann, weil die verantwortliche Rolle entfernt wurde.
- Die Variabilitätspunkte werden so gewählt, dass Rolle und Produkt nur gemeinsam Gegenstand des Tailorings sein können. Damit ist zwar bei Erstellung von Vorgehensmodellinhalten immer noch darauf zu achten, dass auch eine entsprechende Zuordnung zwischen Rolle und Produkt stattfindet, diese Zuordnung kann aber im Rahmen des Tailorings nicht „verloren gehen“.
- Wäre im Beispiel des V-Modell XT auch die Rolle ein Variabilitätspunkt, so wäre eine Lösung, den Prozessingenieur durch Definition einer geeigneten Konsistenzregel im Werkzeug zur Bearbeitung des Vorgehensmodells auf das potenzielle Problem bei der Konfiguration hinzuweisen. In der Fallstudie zum SE Book in Abschnitt 7.2 haben wir basierend auf dem Process Development Environment (PDE) [KKWB10, KKTW10] ein Werkzeug entwickelt, welches eine Vielzahl von Konsistenzregeln während des Entwurfs des Vorgehensmodells, insbesondere beim Entwurf der Konfigurationsanteile, überprüft.

Wenn die strukturelle und inhaltliche Konsistenz nicht konstruktiv sichergestellt werden können, bleibt nur die Prüfung der Bedingungen durch analytische Verfahren. Im Folgenden skizzieren wir exemplarisch ein analytisches Verfahren zur Aufdeckung des in Abbildung 5.11 veranschaulichten strukturellen Widerspruchs.

5.5.2 Analytische Sicherstellung

In folgenden Abschnitt skizzieren wir ein Verfahren zur Prüfung des Vorgehensmodells, bzw. der Konfigurationsanteile, auf Vorhandensein des in Abbildung 5.11 veranschaulichten Konflikts. Dieser Konflikt ist ein typisches Problem des feature-basierten Tailorings und das Verfahren sollte Hinweise für die Lösung vergleichbarer Herausforderungen geben.

Das grobe Verfahren ist in Abbildung 5.13 skizziert.

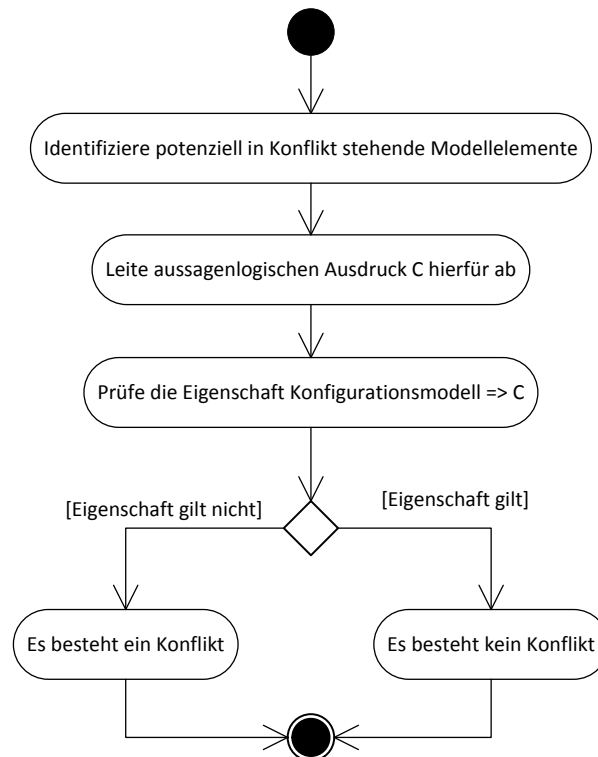


Abbildung 5.13: Analytisches Verfahren zur Erkennung von Konflikten

Zur Erläuterung des Verfahrens vervollständigen wir die Illustration in Abbildung 5.11, so dass wir den aussagenlogischen Ausdruck für das Konfigurationsmodell angeben können. Das Modell, welches zur Erläuterung des Verfahrens dient, ist in Abbildung 5.14 dargestellt.

Die Schritte des Verfahrens aus Abbildung 5.13 im Einzelnen sind:

Identifiziere potenziell in Konflikt stehende Modellelemente: Die Kandidaten für einen Konflikt sind prinzipiell alle variablen Elemente *Element 1*, *Element 2* und *Element 4*. Im Beispiel aus Abbildung 5.14 sind die im Konflikt stehenden variablen Elemente das *Element 1* und das *Element 2*, da sie in einer hierarchischen Beziehung zueinander stehen und an Tailoring-Operationen teilnehmen²: Wenn das Kind-Element *Element 2* Teil der projektspezifischen Ausprägung des Vorgehensmodells sein soll, muss auch das Eltern-Element *Element 1* Teil derselben sein – ansonsten ist die strukturelle Integrität des Vorgehensmodells aufgebrochen.

² *Element 4* ist zwar auch ein Kind von *Element 1*, nimmt aber an keiner Tailoring-Operation teil – in der Modellierung sollte dieses Element möglicherweise besser eine Instanz von *Element* oder *Logisches Element* sein.

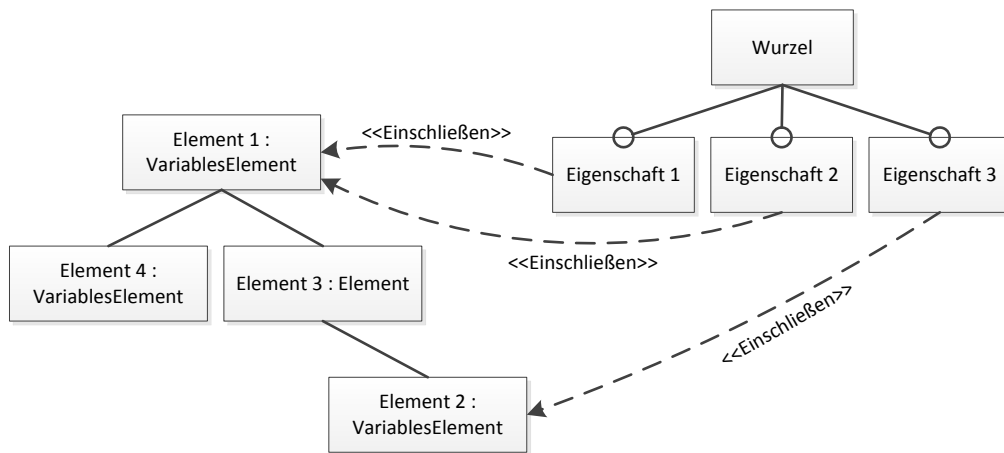


Abbildung 5.14: Beispiel zur Illustration des analytischen Verfahrens zur Erkennung von Konflikten

Leite aussagenlogischen Ausdruck C hierfür ab: Für alle potenziell im Konflikt stehenden Elemente leiten wir einen Ausdruck ab, der gelten muss, damit der Konflikt nicht auftreten kann. Im konkreten Beispiel aus Abbildung 5.14 lautet dieser Ausdruck $Element\ 2 \Rightarrow Element\ 1$. Zur Ableitung eines aussagenlogischen Ausdrucks für die Bedingungen im Vorgehensmodell eignen sich die Regeln für Feature-Modelle aus Tabelle 2.10.

Unter Berücksichtigung der mit $Element\ 1$ und $Element\ 2$ assoziierten Tailoring-Operationen kann man diesen Ausdruck auf Ebene des Konfigurationsmodells „hochziehen“. Im Beispiel aus Abbildung 5.14 lautet der Ausdruck auf Ebene des Konfigurationsmodells:

$$C := Eigenschaft\ 3 \Rightarrow Eigenschaft\ 1 \vee Eigenschaft\ 2$$

Durch etwas Umformung (siehe Tabelle 2.11) ergibt sich:

$$C = \neg Eigenschaft\ 3 \vee Eigenschaft\ 1 \vee Eigenschaft\ 2$$

Prüfe die Eigenschaft $Konfigurationsmodell \Rightarrow C$: Das Konfigurationsmodell stellt selbst bereits einen aussagenlogischen Ausdruck dar. Für das konkrete Beispiel lautet der Ausdruck (zur Ableitung des Ausdrucks sei auf Abschnitt 2.6.2 verwiesen):

$$\begin{aligned} Konfigurationsmodell &:= \\ Eigenschaft\ 1 &\Rightarrow Wurzel \wedge \\ Eigenschaft\ 2 &\Rightarrow Wurzel \wedge \\ Eigenschaft\ 3 &\Rightarrow Wurzel \end{aligned}$$

Auch hier formen wir um und erhalten:

$$\begin{aligned} Konfigurationsmodell &= \\ &(\neg Eigenschaft\ 1 \vee Wurzel) \wedge \\ &(\neg Eigenschaft\ 2 \vee Wurzel) \wedge \\ &(\neg Eigenschaft\ 3 \vee Wurzel) \end{aligned}$$

Wenn die Eigenschaft $Konfigurationsmodell \Rightarrow C$ gilt, so heißt das, die Bedingungen des Konfigurationsmodells sind „stärker“ als durch den Ausdruck C gefordert. Gilt die Eigenschaft nicht, so kann die Bedingung C

5.6 Anwenderunterstützung bei der Durchführung des Tailorings

durch das Konfigurationsmodell nicht sichergestellt werden. In diesem Fall ist eine Konfiguration möglich, welche zum strukturellen Konflikt bei den Vorgehensmodellinhalten führt. Im Beispiel aus Abbildung 5.14 ergibt sich insgesamt die zu prüfende Eigenschaft:

$$\begin{aligned} & (\neg \text{Eigenschaft 1} \vee \text{Wurzel}) \wedge \\ & (\neg \text{Eigenschaft 2} \vee \text{Wurzel}) \wedge \\ & (\neg \text{Eigenschaft 3} \vee \text{Wurzel}) \\ \Rightarrow & \neg \text{Eigenschaft 3} \vee \text{Eigenschaft 1} \vee \text{Eigenschaft 2} \end{aligned}$$

Nach etwas Vereinfachung ergibt sich:

$$\neg \text{Wurzel} \vee \neg \text{Eigenschaft 3} \vee \text{Eigenschaft 1} \vee \text{Eigenschaft 2}$$

Wie zu erwarten, ist diese Bedingung verletzt für *Wurzel* und *Eigenschaft 3* gewählt und *Eigenschaft 1* und *Eigenschaft 2* nicht gewählt.

Es besteht kein Konflikt: Wenn *Konfigurationsmodell* $\Rightarrow C$ gilt, kann der Konflikt aufgrund der Beschaffenheit des Konfigurationsmodells nicht auftreten.

Es besteht ein Konflikt: Wenn die Eigenschaft *Konfigurationsmodell* $\Rightarrow C$ nicht gilt, sollte der Prozessingenieur in geeigneter Weise auf den potenziellen Konflikt hingewiesen werden. Das Verfahren könnte beispielsweise in einem geeigneten Prüf-Werkzeug oder in einem Werkzeug zur Bearbeitung des Vorgehensmodells, wie dem Process Development Environment (PDE), implementiert werden.

Konsistenzbedingungen in realen Vorgehensmodellen hängen stark vom jeweiligen Vorgehensmodell ab. Das hier skizzierte Verfahren sollte aber Anhaltspunkte geben, wie die analytische Sicherstellung von Konsistenz umgesetzt werden kann. Im folgenden Abschnitt befassen wir uns anhand zweier Verfahren mit der Unterstützung für den Anwender des Vorgehensmodells bei der Durchführung des Tailorings.

5.6 Anwenderunterstützung bei der Durchführung des Tailorings

Das feature-basierte Tailoring bietet dem Anwender eines Vorgehensmodells die Möglichkeit, das Vorgehensmodell basierend auf einem potenziell umfangreichen Konfigurationsmodell projektspezifisch anzupassen. Im Vergleich zu alternativen Ansätzen – beispielsweise zum Tailoring-Modell des V-Modell XT (siehe Abschnitt 3.2.1), zum SE Book (vergleiche Abschnitt 3.2.4) oder zu HERMES (vergleiche Abschnitt 3.2.3) – kann das Konfigurationsmodell des feature-basierten Tailorings einen wesentlich größeren Umfang haben. Auf der einen Seite bedeutet das zwar mehr Flexibilität und die Möglichkeit zur feingranularen Anpassung. Auf der anderen Seite steigt dadurch potenziell der Aufwand und die Komplexität der projektspezifischen Anpassung.

In diesem Abschnitt skizzieren wir zwei Verfahren zur Unterstützung des Anwenders bei der (werkzeuggestützten) Anpassung. Beide Verfahren machen Gebrauch von der Semantik der Feature-Modelle und sind auch auf andere Anwendungsbereiche von Featurebäumen übertragbar.

Im ersten Teil dieses Abschnitts skizzieren wir einige Maßnahmen, die den Aufwand und die Komplexität bei der Konfiguration reduzieren können.

Im zweiten Teil stellen wir ein Verfahren zur Ermittlung von Seiteneffekten bei der Konfiguration dar. Gerade die Querbeziehungen (sprich: *Cross-Tree Constraints*) im Konfigurationsmodell erhöhen die Komplexität bei der Konfiguration, da sich eine Änderung an einer Projekteigenschaft auf andere, „weit entfernte“ Bereiche der Konfiguration auswirken kann.

5.6.1 Reduktion des Konfigurationsaufwands

Bei der Umsetzung des feature-basierten Tailorings im SE Book (vergleiche die Fallstudie in Abschnitt 7.2) haben wir einige Mechanismen umgesetzt, die den Aufwand bei der projektspezifischen Anpassung des Vorgehensmodells reduzieren.

Maßnahmen, die den Aufwand und die Komplexität reduzieren können, sind:

Beschreibung zu den Kriterien Jedes Kriterium im SE Book hat eine Beschreibung, die den Sinn des Kriteriums und seine Wirkung auf die Inhalte des Vorgehensmodells angibt.

Visualisierung der Wirkung eines Kriteriums Zu jedem Kriterium im SE Book kann eine detaillierte Aufstellung der durch das Kriterium betroffenen Inhalte des Vorgehensmodells angezeigt werden. Das ist vergleichbar mit der Visualisierung der in die projektspezifische Ausprägung eingeschlossenen Vorgehensbausteine im V-Modell XT (vergleiche z. B. Abbildung 3.4).

Vorbelegung der Konfiguration Im SE Book Modell kann zu jedem Feature eine Standardbelegung angegeben werden. Diese erlaubt dem Anwender, mit einer sinnvollen Ausgangskonfiguration des Vorgehensmodells zu starten. Für die projektspezifische Anpassung muss er nur die Abweichungen von der Vorbelegung angeben. Dadurch wird – auch wenn das gesamte Konfigurationsmodell zur Konfiguration groß ist – der tatsächliche Aufwand bei der projektspezifischen Anpassung deutlich reduziert und die Konfiguration ergibt sofort eine gültige Ausprägung des Vorgehensmodells.

Verfahren zur Bestimmung einer gültigen Konfiguration Der Anwender sollte bei der Konfiguration nicht in eine „Sackgasse“ geraten und eine Änderung an der Konfiguration sollte wieder eine gültige Konfiguration ergeben. In verfügbaren Konfigurationswerkzeugen basierend auf Feature-Modellen, z. B. [KTS⁺09], ist die Wahl eines Features oft unmöglich, da es durch die Belegung eines anderen Features eingeschränkt wird. Klassisches Beispiel sind Features, die in einer XOR-Beziehung zueinander stehen. In einer Gruppe aus XOR-verknüpften Features darf immer nur genau eins gewählt sein. In bestehenden Realisierungen ist oft erst das aktuell gewählte Feature abzuwählen, bevor ein anderes aus der Gruppe gewählt werden kann. Intuitiver wäre hier das Verhalten einer „Radio-Button-Group“.

Im folgenden Abschnitt 5.6.1 stellen wir ein Verfahren dar, welches basierend auf einer gültigen Konfiguration des Feature-Modells und einer Änderung durch den Anwender wieder eine gültige Konfiguration berechnet und damit eine intuitive Bedienung des Konfigurationswerkzeugs erlaubt.

Verfahren zur Bestimmung einer gültigen Konfiguration

Nimmt der Anwender eine Änderung an der Konfiguration vor, sollte das Ergebnis wieder eine gültige Konfiguration im Sinne der durch das Konfigura-

5.6 Anwenderunterstützung bei der Durchführung des Tailorings

tionsmodell definierten Randbedingungen sein. Das in Abbildung 5.15 dargestellte Verfahren berechnet eine derartige Konfiguration abhängig von der Änderung des Anwenders.

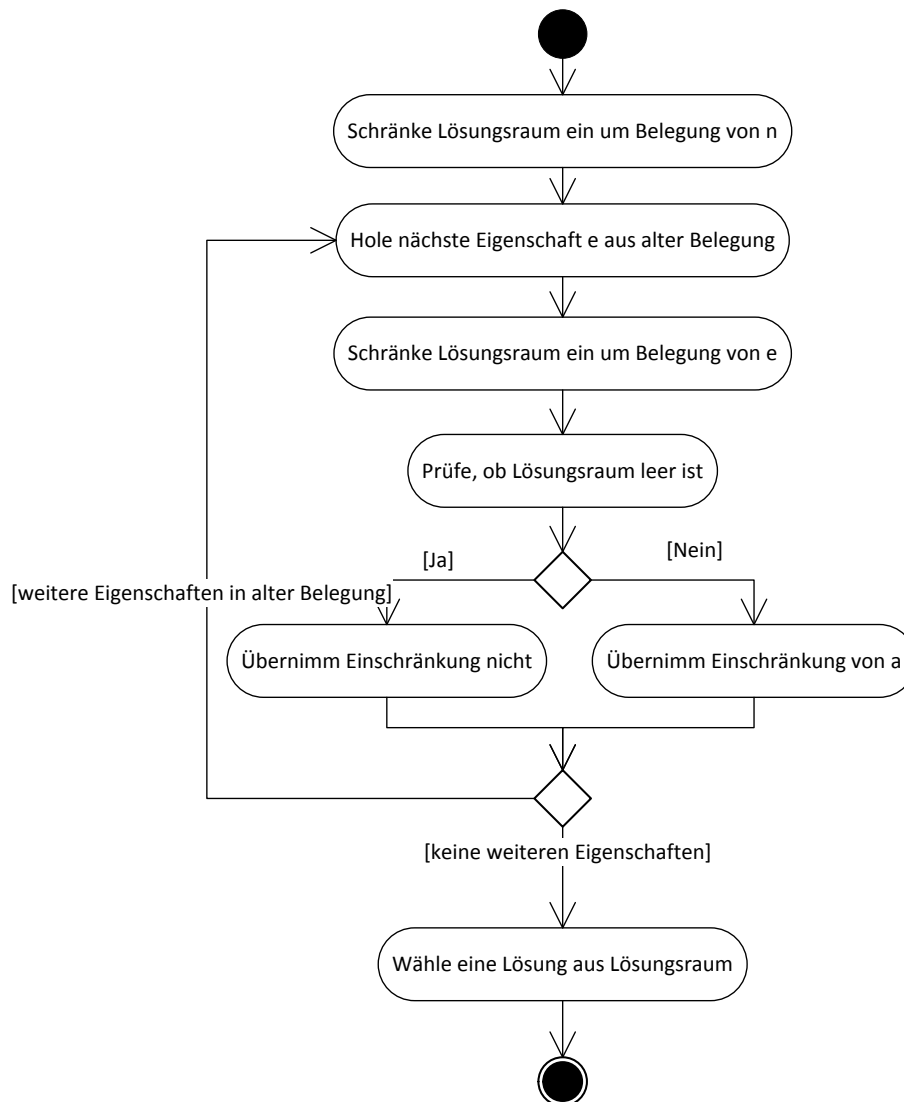


Abbildung 5.15: Verfahren zur Bestimmung einer gültigen Konfiguration

Zur Erläuterung des Verfahrens einige Vorbemerkungen (das Verfahren beruht wesentlich auf der Tatsache, dass das Konfigurationsmodell ein Featurebaum ist):

- Wir bezeichnen als *Belegung* einer Eigenschaft ihren Zustand in der Konfiguration. Eine Eigenschaft kann *gewählt*, *nicht gewählt* oder *nicht belegt* sein. Ist die Eigenschaft *nicht belegt*, so kann die Belegung frei gewählt werden.
- Für das Konfigurationsmodell gibt es eine Menge von Belegungen, welche dieses *erfüllen*. Wir bezeichnen eine das Konfigurationsmodell erfüllende Belegung als *Lösung*.
- Alle möglichen *Lösungen* des Konfigurationsmodells nennen wir *Lösungsraum*.
- Die *alte Belegung* bezeichnet eine konkrete Belegung für alle Eigenschaften des Konfigurationsmodells, d.h. keine Eigenschaft ist *nicht belegt*. Wir

nehmen an, dass die *alte Belegung* eine *Lösung* des Konfigurationsmodells ist.

- Wir können den Lösungsraum einschränken, indem wir für eine Eigenschaft eine Belegung *gewählt* oder *nicht gewählt* fordern.
- Wenn eine Eigenschaft *nicht belegt* ist, kann sie im Lösungsraum beide Belegungen annehmen.

Mit diesen Vorbemerkungen funktioniert das Verfahren wie folgt:

Schränke Lösungsraum ein um Belegung von n : Wir gehen aus von einem komplett uneingeschränkten Lösungsraum. D.h. der Lösungsraum enthält alle möglichen Lösungen für das Konfigurationsmodell. Die Eigenschaft n entspricht der aktuellen Änderung der Konfiguration durch den Anwender. Mit anderen Worten: Er hat die Eigenschaft *gewählt* oder *nicht gewählt*. Da die neue Konfiguration auf jeden Fall die vom Anwender gewünschte Belegung für n enthalten soll, schränken wir den Lösungsraum um die Belegung von n ein.

Hole nächste Eigenschaft e aus alter Belegung: Wir gehen davon aus, dass es eine *alte Belegung* gibt. Diese entspricht der zuletzt gültigen Konfiguration vor der Änderung durch den Anwender. In einer Schleife betrachten wir die Belegung für jede Eigenschaft e im Konfigurationsmodell.

Schränke Lösungsraum ein um Belegung von e : Bei der Betrachtung der ersten Eigenschaft aus der alten Belegung ist der Lösungsraum komplett uneingeschränkt bis auf die Belegung von n . Wir schränken ihn nun für jedes e ein und:

Prüfe, ob Lösungsraum leer ist: Ist der Lösungsraum leer, stehen die Belegungen im Konflikt zueinander. Ist der Lösungsraum nicht leer, gibt es eine Lösung für die Belegung von n und e . Zur Prüfung, ob der Lösungsraum leer ist, kann ein SAT Solver verwendet werden (vergleiche Abschnitt 2.6).

Übernimm Einschränkung von e : Wenn der Lösungsraum nicht leer ist, übernehmen wir die Einschränkung von e .

Übernimm Einschränkung nicht: Ist der Lösungsraum leer, entspricht die Belegung von n und e zusammen nicht einer gültigen Konfiguration. Da die Belegung von n der Änderung des Anwenders entspricht, ist diese Belegung „gesetzt“. Wir lassen e also *nicht belegt* – damit kann e in einer möglichen Lösung sowohl *gewählt* oder *nicht gewählt* werden.

Wähle eine Lösung aus Lösungsraum: Der Lösungsraum ist jetzt eingeschränkt um die Belegung von n und alle Belegungen aus der vorherigen Konfiguration, die nicht mit der Belegung von n im Konflikt stehen. Eigenschaften, deren Belegung mit der Belegung von n im Konflikt standen, sind *nicht belegt* geblieben. Für diese Eigenschaften existiert aber eine Belegung, so dass die gesamte Belegung eine *Lösung* für das Konfigurationsmodell ist. Wir wählen eine derartige Belegung und erhalten eine neue, gültige Konfiguration. Zur Wahl einer Lösung aus dem Lösungsraum kann ein SAT Solver verwendet werden (vergleiche Abschnitt 2.6).

Das hier skizzierte Verfahren berechnet eine neue, gültige Konfiguration basierend auf einer Änderung der Konfiguration durch den Anwender, die sich nur *minimal* von der vorherigen Konfiguration unterscheidet. Da die Belegung der alten Konfiguration soweit als möglich in die neue Konfiguration übernommen wird, unterscheidet diese sich nur in (1) der Änderung des Anwenders und (2) in allen Belegungen, die aufgrund der Änderung des Anwenders „umschalten“ müssen.

5.6 Anwenderunterstützung bei der Durchführung des Tailorings

Wesentliches Merkmal dieses Verfahrens ist, dass die neue Konfiguration – abhängig von der Änderung des Anwenders und den durch das Konfigurationsmodell beschriebenen Randbedingungen an die Konfiguration – abgeleitet wird. Im Gegensatz zu vielen existierenden Konfigurationswerkzeugen basierend auf Feature-Modellen kann der Anwender damit jederzeit jede Eigenschaft ändern und gerät nicht in eine Sackgasse, in der er erst die Belegung einer anderen Eigenschaft ändern müsste, bevor er die eigentlich gewünschte Änderung durchführen kann.

5.6.2 Ermittlung von Seiteneffekten

Die Komplexität der projektspezifischen Konfiguration wird für den Anwender entscheidend dadurch erhöht, dass eine Änderung der Konfiguration Seiteneffekte auf „weit entfernte“ Bereiche im Konfigurationsmodell haben kann. Derartige Seiteneffekte entstehen, wenn das Konfigurationsmodell Querbeziehungen zwischen Projekteigenschaften enthält. Zum Verständnis der Konsequenzen seiner Konfiguration ist es für den Anwender entscheidend, dass er Kenntnis derartiger Seiteneffekte hat.

Im Folgenden skizzieren wir ein mögliches Verfahren, den Anwender während der Konfiguration auf Seiteneffekte resultierend aus Querbeziehungen (sprich: *Cross-Tree Constraints*) hinzuweisen.

Wir nehmen die Existenz folgender Hilfsalgorithmen an:

- *BerechneGueltigeKonfiguration*: dieser Algorithmus berechnet basierend auf der Änderung der Konfiguration durch den Anwender eine neue, gültige Konfiguration. Dieser Algorithmus entspricht dem in Abschnitt 5.6.1 skizzierten Verfahren.
- *BerechneGueltigeKonfigurationOhneCTC*: dieser Algorithmus berechnet ebenfalls eine neue, gültige Konfiguration – allerdings berücksichtigt er die Cross-Tree Constraints im Konfigurationsmodell nicht. Durch Nichtberücksichtigung der Cross-Tree Constraints erhält man ein „schwächeres“ Modell. Im Umkehrschluss heißt das, dass die möglichen Lösungen eine Obermenge der Lösungen mit Berücksichtigung der Cross-Tree Constraints sind.
- *BerechneGeaenderteEigenschaften*: dieser Algorithmus berechnet das „Delta“ zwischen einer alten und einer neuen Konfiguration. Er ermittelt alle Projekteigenschaften, deren Belegung sich zwischen beiden Konfigurationen unterscheidet.

Mit diesen Hilfsalgorithmen lassen sich die Seiteneffekte einer Änderung an der Konfiguration mit dem Algorithmus 5.6.1 ermitteln.

Algorithm 5.6.1: ERMITTELSEITENEFFEKTE()

```
a ← BERECHNEGUELTIGEKONFIGURATION()
b ← BERECHNEGUELTIGEKONFIGURATIONOHNECTC()
fa ← BERECHNEGEAENDERTEEIGENSCHAFTEN(a)
fb ← BERECHNEGEAENDERTEEIGENSCHAFTEN(b)
if fa \ fb = ∅
  then output (EsGibtKeineSeiteneffekte)
  else output (EsGibtSeiteneffekte)
```

Der Algorithmus stützt sich ab auf die Tatsache, dass die Menge der geänderten Projekteigenschaften *minimal* ist, wenn sie mit dem Verfahren aus Abschnitt 5.6.1 berechnet wird. Die Menge *fa* enthält *alle* geänderten Projekteigenschaften. Die Menge *fb* enthält nur diejenigen Eigenschaften, die nicht aufgrund von Querbeziehungen geändert wurden. Wenn die Differenz zwischen *fa* und *fb* nicht leer ist, heißt das: es gibt Belegungen, die aufgrund von Querbeziehungen geändert werden mussten und es sollte ein Hinweis an den Anwender mit der Anzeige der geänderten Projekteigenschaften erfolgen.

5.7 Zusammenfassung

In diesem Kapitel haben wir die Strukturen für das feature-basierte Tailoring von Vorgehensmodellen entwickelt und das Verhalten des feature-basierten Tailorings – insbesondere im Zusammenspiel zwischen Konfigurationsmodell und Vorgehensmodellinhalten – festgelegt.

Im ersten Teil des Kapitels haben wir uns mit der Situationsbeschreibung eines Projekts anhand eines Katalogs von Projektcharakteristika befasst und ein Modell für die Beschreibung der Charakterisierung, basierend auf der Technik der Feature-Bäume, abgeleitet.

Im zweiten Teil des Kapitels haben wir für mögliche Tailoring-Operationen eine Semantik entwickelt und daraus eine Grundoperation auf den Vorgehensmodellinhalten abgeleitet.

Die Synthese des Konfigurationsmodells, der Tailoring-Operationen und der Inhalte des Vorgehensmodells zum Gesamtmodell des feature-basierten Tailorings war Gegenstand des dritten Teils dieses Kapitels. Um unabhängig von einem konkreten Vorgehensmodell zu bleiben, haben wir uns bei der Beschreibung der Vorgehensmodellinhalte auf das einfache Meta-Metamodell für Vorgehensmodelle aus Kapitel 4 zurückgezogen.

In den Abschnitten 5.5 und 5.6 haben wir einige Strategien und Verfahren zur Sicherstellung der Konsistenz der projektspezifischen Ausprägung des Vorgehensmodells und zur Unterstützung des Vorgehensmodell-Anwenders bei der Durchführung des Tailorings skizziert. Konkrete Randbedingungen an das Tailoring hängen stark vom jeweiligen Vorgehensmodell ab. Die skizzierten Verfahren sollten aber einen Eindruck geben, wie die in diesem Kapitel entwickelten Konzepte genutzt werden können.

Anwendung des feature-basierten Tailorings

In Kapitel 4 haben wir die Struktur von Vorgehensmodellen analysiert und die Grundlage für den Ansatz des feature-basierten Tailorings gelegt. In Kapitel 5 haben wir die konzeptionellen und technischen Mechanismen des Ansatzes entwickelt. In diesem Kapitel widmen wir uns der Umsetzung und Anwendung des feature-basierten Tailorings.

Bei der Umsetzung gehen wir davon aus, dass ein bestehendes Vorgehensmodell um die Fähigkeiten des feature-basierten Tailorings erweitert werden soll. Dementsprechend nehmen wir die Existenz von Strukturen des Vorgehensmodells (in Form eines Metamodells) und von Vorgehensmodellinhalten an.

Am Ende dieses Kapitels wird der Leser die Theorie zum feature-basierte Tailoring aus Kapitel 5 in einem existierenden Vorgehensmodell umsetzen und das Tailoring eines Vorgehensmodells basierend auf dem in dieser Arbeit entwickelten Ansatz durchführen können.

Übersicht

6.1	Überblick	146
6.2	Rollen	146
6.3	Feature-basiertes Tailoring implementieren	147
6.4	Vorgehensmodell projektspezifisch anpassen	160
6.5	Beispiele	161
6.6	Zusammenfassung	168

6.1 Überblick

Bei der Anwendung des feature-basierten Tailorings unterscheiden wir zwei Perspektiven:

- Der Prozessingenieur oder Vorgehensmodell-Entwickler implementiert das feature-basierte Tailoring im Vorgehensmodell, definiert Variabilitätspunkte und Tailoring-Kriterien und stellt das Vorgehensmodell dem Vorgehensmodell-Anwender zur Verfügung.
- Der Anwender eines Vorgehensmodells wiederum führt die projektspezifische Anpassung des Vorgehensmodells für sein konkretes Projekt durch indem er die vom Prozessingenieur definierten Kriterien entsprechend der Charakteristika seines Projekts mit Werten belegt.

In Anlehnung an [Kuh07] sind je nach Perspektive die in Abbildung 6.1 dargestellten Anwendungsfälle relevant.

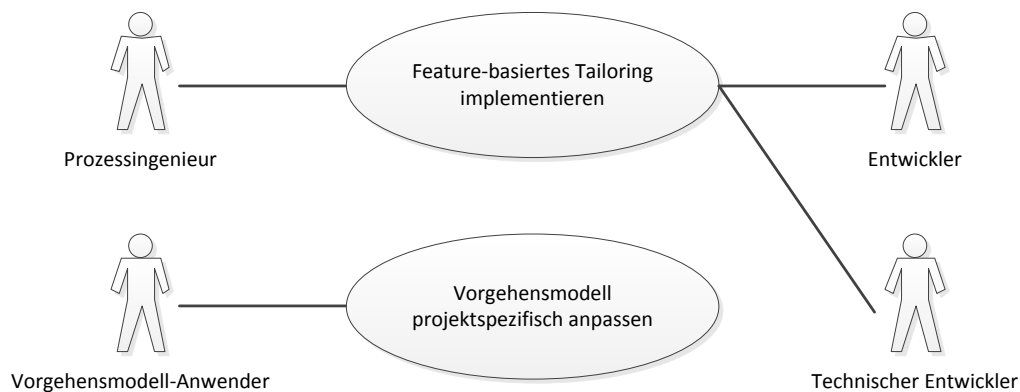


Abbildung 6.1: Anwendungsfälle für die Umsetzung und Anwendung des feature-basierten Tailorings (angelehnt an [Kuh07])

In diesem Kapitel beschreiben wir die zur Umsetzung und Anwendung des feature-basierten Tailorings relevanten Anwendungsfälle aus Abbildung 6.1. Nach einem Überblick über die beteiligten Rollen gehen wir in Abschnitt 6.3 detailliert auf die Umsetzung des feature-basierten Tailorings ein und beleuchten in Abschnitt 6.4 das feature-basierte Tailoring aus Sicht des Vorgehensmodell-Anwenders. Den Abschluss dieses Kapitels bildet in Abschnitt 6.5 eine Sammlung von Umsetzungsbeispielen.

6.2 Rollen

Die an den Anwendungsfällen beteiligten Rollen sind:

Prozessingenieur: Der Prozessingenieur ist verantwortlich für die Entwicklung und Pflege eines Vorgehensmodells. Er hat fundierte Kenntnis der inneren Struktur des Vorgehensmodells, seiner Inhalte und der zur Bearbeitung des Vorgehensmodells verwendeten Werkzeuge. Er ist in der Lage, Anforderungen an das Vorgehensmodell in den gegebenen Strukturen abzubilden und umzusetzen.

Entwickler: Nicht alle inhaltlichen Arbeiten sind vom Prozessingenieur selbst durchzuführen. Insbesondere Rückmeldungen aus der Anwendung des

Vorgehensmodells können wertvolle Beiträge zur kontinuierlichen Verbesserung bringen. Unter Entwickler in diesem Zusammenhang verstehen wir einen Entwickler für Inhalte des Vorgehensmodells. Dies können beispielsweise Vertreter von Interessensgruppen innerhalb der Organisation und die Geschäftsleitung sein. Insbesondere ist in dieser Rolle auch die inhaltliche Mitwirkung der Projektteams am Vorgehensmodell aufgehoben – beispielsweise als Rückmeldung aus der Anwendung in einem Projekt.

Technischer Entwickler: Das feature-basierte Tailoring kann in einem Vorgehensmodell nur umgesetzt werden, wenn das Vorgehensmodell durch geeignete Werkzeuge begleitet wird. Der technische Entwickler hat ein Verständnis der inneren Struktur und der Konzepte des Vorgehensmodells und die technischen Kompetenzen zur Realisierung und Erweiterung der Werkzeuge zum Vorgehensmodell.

Anwender: Der Anwender des Vorgehensmodells führt die projektspezifische Anpassung zur Erstellung einer projektspezifischen Instanz des Vorgehensmodells bei der Projektinitialisierung durch. Diese Rolle wird üblicherweise vom Projektleiter eingenommen. Zwar sind alle Akteure im Projekt potenziell Anwender des Vorgehensmodells, nicht aber des Tailorings. Da die Verantwortung für das Vorgehen im Projekt üblicherweise beim Projektleiter liegt, fällt ihm auch die Aufgabe der Durchführung des Tailorings zu. Im Gegensatz zum Prozessingenieur gehen wir beim Anwender nicht davon aus, dass er detaillierte Kenntnis der inneren Struktur des Vorgehensmodells hat.

6.3 Feature-basiertes Tailoring implementieren

Der Anwendungsfall *Feature-basiertes Tailoring implementieren* fasst die Tätigkeiten zur Umsetzung des feature-basierten Tailorings in einem Vorgehensmodell zusammen. Er setzt sich zusammen aus verfeinerten Anwendungsfällen für die Einzelaufgaben. Ein Überblick über den Anwendungsfall *Feature-basiertes Tailoring implementieren* und seine verfeinerten Teil-Anwendungsfälle ist in Abbildung 6.2 dargestellt.

Akteur: Prozessingenieur, Entwickler, Technischer Entwickler

Auslöser: (1) An ein bestehendes Vorgehensmodell besteht die Anforderung zur Flexibilisierung der projektspezifischen Anpassung mit Hilfe des feature-basierten Tailorings. (2) Ein Vorgehensmodell soll neu entworfen werden und dabei die Konzepte und Mechanismen des feature-basierten Tailorings beinhalten.

Ergebnis: Es liegen die (Meta-)Modellstrukturen zur Abbildung des feature-basierten Tailorings, ein Kriterienkatalog zur projektspezifischen Anpassung in Form des Konfigurationsmodells, projektspezifisch anpassbare Vorgehensmodellinhalte und ein Werkzeug zur Durchführung der projektspezifischen Anpassung vor.

Vorbedingungen: Es ist (gegebenenfalls) ein Metamodell verfügbar, das den technischen Rahmen vorgibt. Es sind gegebenenfalls Vorgehensmodellinhalte, die Gegenstand der projektspezifischen Anpassung werden sollen, verfügbar. Es sind Anforderungen für die Neuentwicklung/Aktualisierung verfügbar.

6.3 Feature-basiertes Tailoring implementieren

Nachbedingungen: Das feature-basierte Tailoring ist im Vorgehensmodell umgesetzt. Alle Anforderungen beziehungsweise Änderungsanforderungen sind in Form von neuen/aktualisierten Vorgehensmodellinhalten umgesetzt. Gegebenenfalls steht ein Werkzeug zur Durchführung der projektspezifischen Anpassung basierend auf dem feature-basierten Tailoring zur Verfügung.

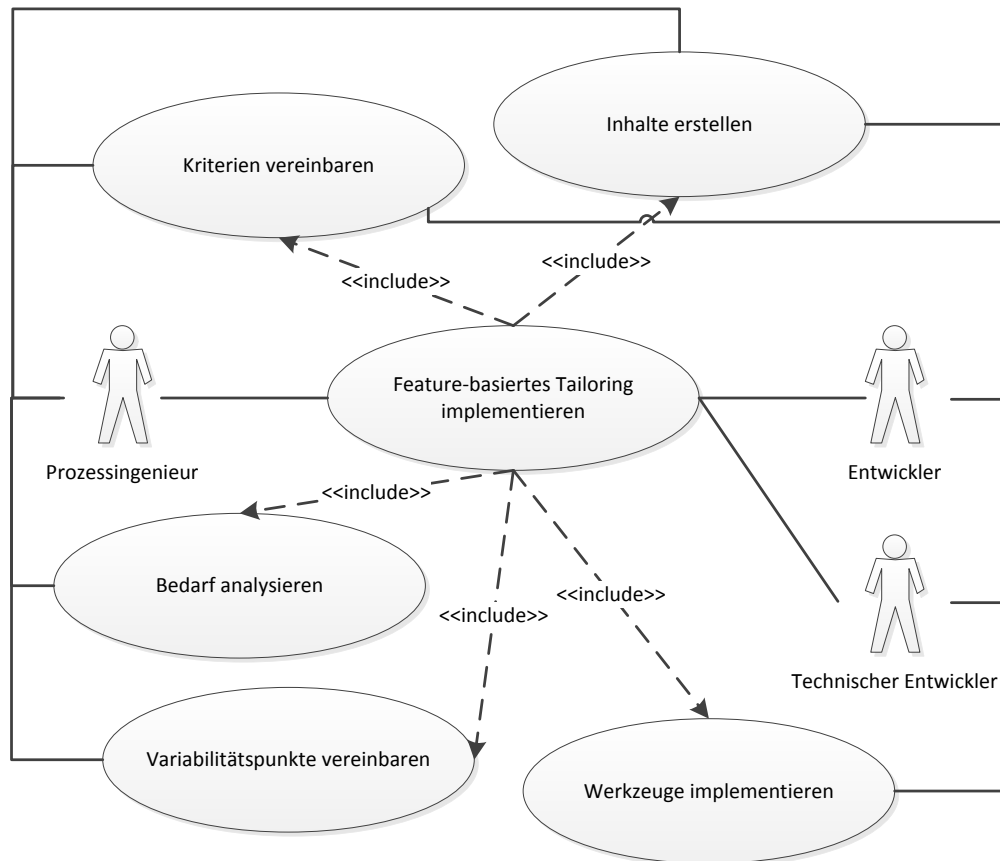


Abbildung 6.2: Anwendungsfälle für die Implementierung des feature-basierten Tailorings

Voraussetzung für die Umsetzung des feature-basierten Tailorings ist ein Vorgehensmodell, dem ein Metamodell unterliegt. In der Regel wird ein derartiges Vorgehensmodell bereits existieren. Wird ein Vorgehensmodell (und sein Metamodell) komplett neu entwickelt, können die Modelle aus Kapitel 5 verwendet werden, um die Strukturen für das feature-basierte Tailoring darin zu verankern.

6.3.1 Produkte

Im Anwendungsfall *Feature-basiertes Tailoring implementieren* werden die folgenden Produkte neu erstellt oder bearbeitet:

Umsetzungskonzept Das Umsetzungskonzept ist das Ergebnis des Anwendungsfalls *Bedarf analysieren* (siehe Abschnitt 6.3.3).

Voraussetzung: keine benötigten Produkte.

Metamodell Das Metamodell wird im Anwendungsfall *Variabilitätspunkte vereinbaren* bearbeitet (siehe Abschnitt 6.3.4). Der Anwendungsfall beschränkt sich auf die Definition der Variabilitätspunkte im Metamodell. Die grundlegenden Strukturen des Vorgehensmodells werden als gegeben angenommen.

Voraussetzung: Umsetzungskonzept, Metamodell ohne Strukturen für feature-basiertes Tailoring.

Konfigurationsmodell Das Konfigurationsmodell wird im Anwendungsfall *Kriterien vereinbaren* (siehe Abschnitt 6.3.5) erstellt oder bearbeitet. Es besteht aus einem Katalog von Kriterien zur Charakterisierung der Projekte und einem Feature-Modell, welches den Kriterienkatalog strukturiert, Kriterien miteinander in Beziehung setzt und die Kriterien über die Tailoring-Operationen mit den Vorgehensmodellinhalten verknüpft.

Voraussetzung: Umsetzungskonzept, Metamodell.

Variabler Inhalt Variable Inhalte für das Vorgehensmodell werden im Anwendungsfall *Inhalte erstellen* (siehe Abschnitt 6.3.6) neu erstellt und bearbeitet. Der Anwendungsfall *Inhalte erstellen* bezieht sich in erster Linie auf diejenigen Inhalte des Vorgehensmodells, die Gegenstand der projektspezifischen Anpassung sein können. Andere, in jeder projektspezifischen Ausprägung vorhandene Inhalte (*Commonalities*) nehmen wir als gegeben an.

Voraussetzung: Umsetzungskonzept, Metamodell, Konfigurationsmodell.

Tailoring-Werkzeug Das Tailoring-Werkzeug wird im Anwendungsfall *Werkzeuge implementieren* (siehe Abschnitt 6.3.7) neu entwickelt oder bearbeitet. Es stützt sich auf das Metamodell des Vorgehensmodells ab, insbesondere auf das Konfigurationsmodell und die Variabilitätspunkte.

Voraussetzung: Umsetzungskonzept, Metamodell.

Der Zusammenhang der Produkte und ihre Beziehung zu den Anwendungsfällen ist in Abbildung 6.3 dargestellt.

6.3.2 Ablauf

In Abbildung 6.3 ist ein möglicher Ablauf zum Anwendungsfall *Feature-basiertes Tailoring implementieren* veranschaulicht¹: nach der Analyse der Anforderungen liegt ein Umsetzungskonzept mit einer Liste von Anforderungen vor.

Mit dem Umsetzungskonzept steht fest, ob das Metamodell des Vorgehensmodells betroffen ist, insbesondere weil die Variabilitätspunkte erstmalig zu definieren sind oder Änderungsbedarf an den Variabilitätspunkten besteht. Eine Änderung am Metamodell zieht in der Regel eine Anpassung des Werkzeugs zur Durchführung des Tailorings nach sich.

Weiterhin steht mit dem Umsetzungskonzept fest, ob die Kriterien zur Projektcharakterisierung neu erstellt oder überarbeitet werden müssen. Eine Änderung an der Charakterisierung zieht in der Regel eine Nacharbeit bei den betroffenen variablen Inhalten nach sich. Ergibt sich aus dem Umsetzungskonzept kein Bedarf für eine Bearbeitung der Projektcharakterisierung, besteht der Anwendungsfall *Feature-basiertes Tailoring implementieren* nur aus einer Bearbeitung der Inhalte des Vorgehensmodells und damit dem Anwendungsfall *Inhalte erstellen* (siehe Abschnitt 6.3.6).

¹ Aus Gründen der Übersichtlichkeit sind die Verbindungen zwischen dem Umsetzungskonzept bzw. den Anforderungen und den anderen Produkten nicht eingezeichnet: Das Umsetzungskonzept gibt die Anforderungen für die anderen Produkte vor.

6.3 Feature-basiertes Tailoring implementieren

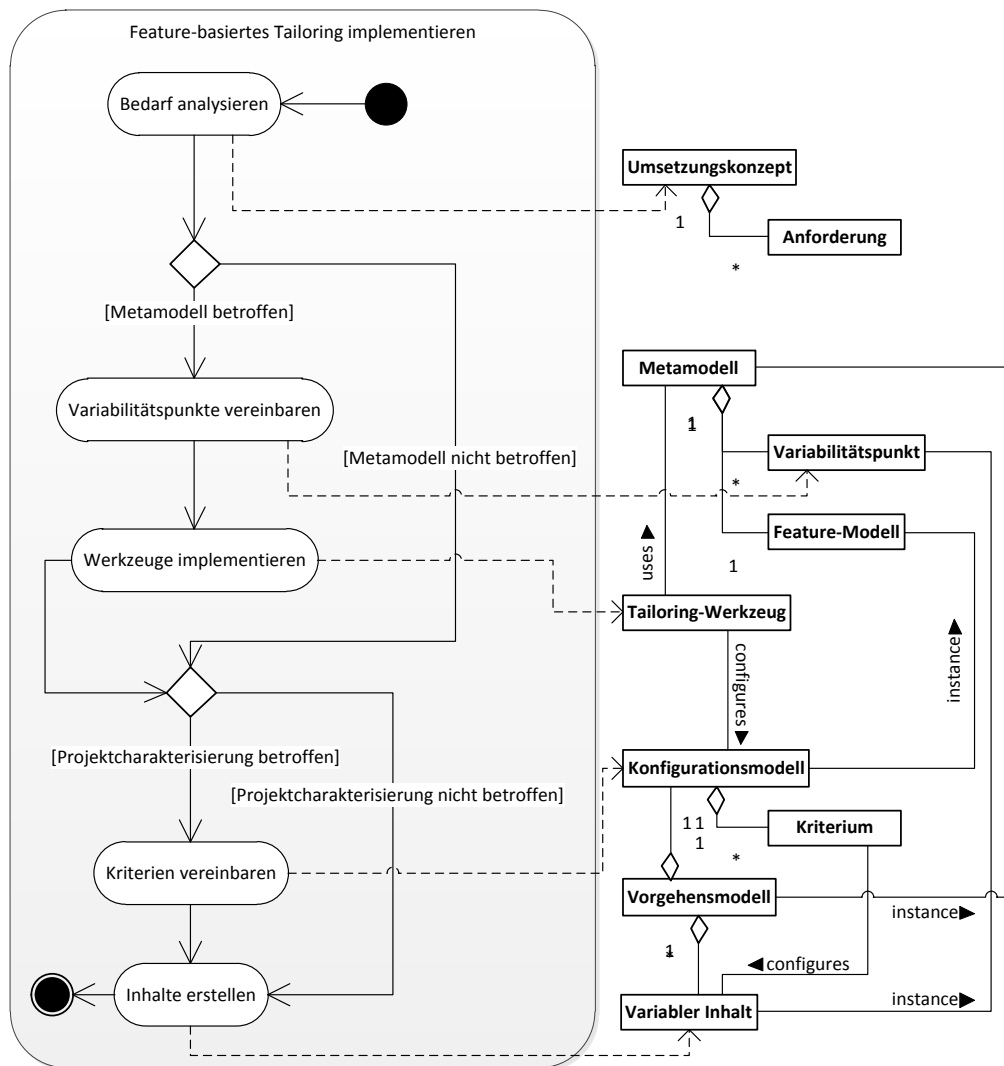


Abbildung 6.3: Prozess: Feature-basiertes Tailoring implementieren

6.3.3 Anwendungsfall: Bedarf analysieren

Zu Beginn der Neuentwicklung oder Änderung des feature-basierten Tailorings steht eine Analyse der Anforderungen.

Das Ergebnis der Bedarfsanalyse ist ein Umsetzungskonzept und ein Katalog von Anforderungen an das feature-basierte Tailoring. Abhängig davon, ob das feature-basierte Tailoring zum ersten Mal in einem Vorgehensmodell umgesetzt wird (inklusive der Entwicklung eines Werkzeugs zur Durchführung des Tailorings), oder ob es bereits umgesetzt ist und Inhalte (und Kriterien) zu erstellen oder anzupassen sind, fällt der Umfang des Umsetzungskonzepts und des Anforderungskatalogs unterschiedlich aus.

Akteur: Prozessingenieur

Auslöser: Das feature-basierte Tailoring ist in einem Vorgehensmodell umzusetzen oder anzupassen.

Ergebnis: Es liegt ein Umsetzungskonzept und ein Katalog der Anforderungen für die Umsetzung oder Anpassung vor.

Vorbedingungen: Die Interessensgruppen am Vorgehensmodell sind verfügbar und willens, an der Anforderungserhebung mitzuwirken.

Nachbedingungen: Das Umsetzungskonzept und ein Katalog der Anforderungen an die Entwicklung oder Anpassung des feature-basierten Tailorings im Vorgehensmodell liegen vor und sind von den Interessensgruppen abgesegnet.

Zur Ermittlung der Anforderungen an das feature-basierte Tailoring eignet sich ein Verfahren, welches sich an die in Abschnitt 2.4.3 beschriebene Studie anlehnt.

Wenn das Vorgehensmodell bereits im Einsatz ist und um die Fähigkeit zur projektspezifischen Anpassung mittels des feature-basierten Tailorings erweitert werden soll, sollten alle verfügbaren Unterlagen und Erfahrungsberichte zum Vorgehensmodell und dessen Einsatz ausgewertet werden.

Zur Definition geeigneter Variabilitätspunkte, zur Vereinbarung des die Projekte charakterisierenden Kriterienkatalogs (und damit des Konfigurationsmodells) und zur Erstellung variabler Inhalte ist die Rückmeldung der Anwender des Vorgehensmodells erforderlich. Um diese Rückmeldung einzuholen, kommen beispielsweise Fragebogen, Interviews und Workshops in Frage. Die Tabellen 2.3 und 2.4 eignen sich als Orientierungshilfe für die Entwicklung eines ausführlichen Fragebogens.

Als Vorschlag für einen schlanken Interviewleitfaden übernehmen wir die im Rahmen des *Methodology Shaping* empfohlenen Fragen aus [Coc04]:

1. Können Sie ein Beispielexemplar für jeden Artefakttyp zeigen?
2. Können Sie kurz die Geschichte des Projekts wiedergeben?
3. Was lief im Projekt nicht optimal und was sollte auf keinen Fall wiederholt werden?
4. Was lief gut im Projekt und sollte auf jeden Fall in zukünftige Projekte übernommen werden?
5. Was sind die Prioritäten in Bezug auf die positiven Eigenschaften des Projekts? Was ist am wichtigsten und was wäre verhandelbar?
6. Haben Sie weitere Kommentare oder Bemerkungen?

6.3 Feature-basiertes Tailoring implementieren

Die Interviews zum *Methodology Shaping* in Crystal Clear sollen mit möglichst vielen Kollegen unterschiedlicher Rollen und in unterschiedlichen Teams durchgeführt werden, um ein Bild über Stärken, Schwächen und beherrschende Themen in der Organisation zu bekommen. Diese Sammlung an Informationen dient dann als Grundlage zur Formung einer Vorgehensweise.

Soll das Vorgehensmodell neu entwickelt werden, empfiehlt es sich, bei der Konzipierung etablierte Standards und Metamodelle, wie beispielsweise das V-Modell XT [RHB⁺06], das Metamodel SPem [OMG08] und darauf aufbauend EPF [Ecl11a], zu berücksichtigen.

6.3.4 Anwendungsfall: Variabilitätspunkte vereinbaren

Die Wahl und Definition der Variabilitätspunkte im Vorgehensmodell erfordert fundierte Kenntnis der inneren Struktur des Vorgehensmodells, seines Metamodels und die Fähigkeit, das Metamodel zu erweitern – insbesondere den sicheren Umgang mit den dabei zum Einsatz kommenden Werkzeugen.

Die Aufgabe der Vereinbarung von Variabilitätspunkten fällt dem Prozessingenieur zu.

Die Vereinbarung der Variabilitätspunkte sollte in der Regel nur bei der initialen Umsetzung des feature-basierten Tailorings in einem Vorgehensmodell notwendig sein. Während des Lebenszyklus' des Vorgehensmodells wird dessen innere Struktur in der Regel (weitgehend) unverändert bleiben.

In den meisten Fällen zieht eine Änderung der Variabilitätspunkte eine entsprechende Anpassung des Werkzeugs zur Durchführung des Tailorings nach sich.

Akteur: Prozessingenieur

Auslöser: (1) Das feature-basierte Tailoring soll initial in einem Vorgehensmodell umgesetzt werden. (2) Die definierten Variabilitätspunkte haben sich bei der Anwendung des feature-basierten Tailorings als ungeeignet herausgestellt und müssen angepasst werden.

Ergebnis: Das Metamodel des Vorgehensmodells mit definierten „Angriffspunkten“ für die Tailoring-Operationen. Wird das feature-basierte Tailoring initial in einem Vorgehensmodell umgesetzt, ist ein weiteres Ergebnis das Konfigurationsmodell und die Verknüpfung desselben mit den Variabilitätspunkten über die Tailoring-Operationen.

Vorbedingungen: Das Metamodel zum Vorgehensmodell liegt vor. Auch bei einer vollständigen Neuentwicklung des Vorgehensmodells und seines Metamodels nehmen wir für diesen Anwendungsfall an, dass die grundlegenden Strukturen des Vorgehensmodells bereits definiert sind.

Nachbedingungen: Das Metamodel des Vorgehensmodells enthält eine definierte Menge an Variabilitätspunkten, also Modellelementen, die mit dem Konfigurationsmodell über die Tailoring-Operationen verbunden sind. Vorher existierende Inhalte des Vorgehensmodells sind auf die neuen Strukturen migriert.

Welche Elemente im Vorgehensmodell Angriffspunkt für das Tailoring sein können, bestimmt maßgeblich die Komplexität bei der Entwicklung und Pflege des Vorgehensmodells. Grob kann man bei der Wahl geeigneter Variabilitätspunkte zwischen zwei Extremen unterscheiden: auf der linken Seite von Abbildung 6.4 sind die Variabilitätspunkte so gewählt, dass sich deren Inhalte nicht „überlappen“ und dass Variabilitätspunkte sich über die Hierarchie der Vorgehensmodellinhalte nicht gegenseitig enthalten. Auf der rechten Seite von Abbildung

6.4 überschneiden sich die Inhalte der Variabilitätspunkte und die Variabilitätspunkte sind über die Hierarchie der Vorgehensmodellinhalte verschachtelt.

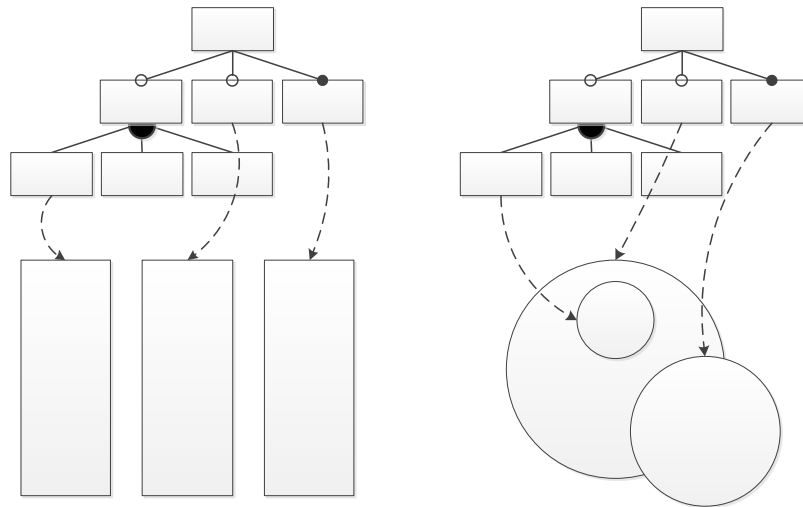


Abbildung 6.4: Extreme bei der Wahl der Variabilitätspunkte

Im linken Fall von Abbildung 6.4 gilt insbesondere die Eigenschaft:

$$\forall ve \in VariableElemente : Kinder(ve) \cap VariableElemente = \emptyset$$

Ist diese Eigenschaft erfüllt, kann es nicht zu strukturellen Konflikten in der projektspezifischen Ausprägung des Vorgehensmodells aufgrund des Tailorings kommen. Kann die Eigenschaft nicht erfüllt werden, muss die strukturelle Integrität der projektspezifischen Ausprägung analytisch sichergestellt werden. In Abschnitt 5.5 haben wir strukturelle und inhaltliche Konsistenz diskutiert und Verfahren zur Sicherstellung skizziert.

Für die Wahl geeigneter Variabilitätspunkte sollten – basierend auf der Analyse aus Abschnitt 6.3.3 – folgende Fragen beantwortet werden:

- Welche Inhalte im Vorgehensmodell müssen variabel sein?
- Welchen Modellelementen entsprechen diese Inhalte?
- Gibt es Modellelemente, die aufgrund ihres Inhalts oder aufgrund von Konventionen des Vorgehensmodells „zusammengehören“?
- Gibt es „Container“-Elemente für die variablen Inhalte bzw. Elemente?
- Lassen sich gegebenenfalls geeignete Container-Elemente definieren?

Ein Beispiel für ein „Container“-Element für inhaltlich zusammengehörige Elemente des Vorgehensmodells ist der *Vorgehensbaustein* im V-Modell XT (vergleiche auch die Fallstudie zum V-Modell XT in Abschnitt 7.3).

Wenn möglich (und inhaltlich gerechtfertigt) ist ein „Schnitt“ der Variabilitätspunkte wie im linken Teil der Abbildung 6.4 zu bevorzugen. Allerdings lassen sich die Anforderungen an das Tailoring nicht immer in dieser Weise darstellen. Ist mehr Flexibilität in der Variabilität des Vorgehensmodells erforderlich, sollten die Verfahren aus Abschnitt 5.5 zur Anwendung kommen, um eine strukturell und inhaltlich sinnvolle projektspezifische Ausprägung zu gewährleisten.

6.3 Feature-basiertes Tailoring implementieren

Wenn das feature-basierte Tailoring in einem Vorgehensmodell überhaupt erst realisiert werden soll, müssen neben der Vereinbarung der Variabilitätspunkte auch die Metamodell-Strukturen für das Konfigurationsmodell zur Beschreibung der Projektcharakteristika geschaffen werden. Ein allgemeines Metamodell für das Konfigurationsmodell findet sich in Abbildung 5.4. Beispiele für konkrete Umsetzungen dieses Modells finden sich in Abbildung 7.4 und 7.13. Diese sollten für ein konkretes Vorgehensmodell adaptierbar sein.

Eine Änderung am Metamodell des Vorgehensmodells erfordert eine Migration der Vorgehensmodellinhalte von der alten auf die neue Struktur. Für die meisten Änderungen am Vorgehensmodell sollte es möglich sein, die Inhalte automatisiert (durch geeignete Transformation) von der alten auf die neue Struktur zu migrieren.

6.3.5 Anwendungsfall: Kriterien vereinbaren

Ausgehend von den Anforderungen an die Variabilität des Vorgehensmodells aus Abschnitt 6.3.3 und den Festlegungen hinsichtlich der variablen Modellanteile aus Abschnitt 6.3.4 werden geeignete Kriterien zur Beschreibung der Variabilität vereinbart und im Konfigurationsmodell abgebildet. Der Kriterienkatalog und die geeignete Abbildung ins Konfigurationsmodell wird vom Prozessingenieur in Zusammenarbeit mit inhaltlichen Entwicklern erarbeitet.

Akteur: Prozessingenieur, Entwickler

Auslöser: (1) Das feature-basierte Tailoring wird in einem Vorgehensmodell initial umgesetzt. (2) Bei Anwendung des feature-basierten Tailorings hat sich der Bedarf an einer Überarbeitung oder Weiterentwicklung der Kriterien zur Charakterisierung der Projekte ergeben.

Ergebnis: Es liegt ein Katalog mit Kriterien zur Charakterisierung der Projekte vor. Die Kriterien sind alle durch geschlossene Fragen zu beantworten und im Konfigurationsmodell umgesetzt.

Vorbedingungen: Die Metamodelle zum Vorgehensmodell und zum Konfigurationsmodell liegen vor. Die inhaltlichen Anforderungen an die Variabilität des Vorgehensmodells bei der projektspezifischen Anpassung liegen vor.

Nachbedingungen: Der Kriterienkatalog zur Projektcharakterisierung ist im Konfigurationsmodell umgesetzt und mit den betroffenen Vorgehensmodellinhalten in Beziehung gesetzt.

Als Ausgangsbasis für mögliche Kriterien zur Charakterisierung der Projekte kann der Katalog in Anhang A verwendet werden. Anstatt von den Kriterien auszugehen, kann auch von den Anforderungen an die Inhalte beim Tailoring ausgegangen werden: Welche Inhalte sollen variabel sein? Welche Inhalte können bei der projektspezifischen Anpassung alternativ gewählt werden? Welche Kriterien lassen sich finden, um die Alternativen zu charakterisieren?

Bei der Wahl der Kriterien zur Charakterisierung von Projekten sollten folgende Punkte beachtet werden:

- In Bezug auf die Anzahl der Kriterien muss abgewogen werden zwischen einer ausreichenden Menge zur sinnvollen Charakterisierung und der Komplexität bei der Durchführung der Anpassung durch zu viele Kriterien.
- Nicht alle Kriterien sind eindeutig und objektiv zu beantworten. Die Einschätzung der Kompetenz der verfügbaren Endanwender beispielsweise

ist möglicherweise gar nicht oder nur subjektiv zu treffen. Das Tailoring basierend auf subjektiver Einschätzung der Projektsituation führt zu einer dementsprechend subjektiv beeinflussten projektspezifischen Ausprägung des Vorgehensmodells.

- Kriterien machen nur dann Sinn, wenn sie auch eine Wirkung auf das Tailoring-Ergebnis haben. In der Literatur werden als Projektcharakteristika z. B. die Faktoren *Fluktuation im Team* und *Zusammenarbeit des Teams in früheren Projekten* genannt (vergleiche Kriterienkatalog im Anhang A). Beide Faktoren beeinflussen den Bedarf an Dokumentation und Kommunikation, z. B. in der Form formalisierter Besprechungen. Es ist allerdings fraglich, ob die beiden Kriterien eine *unterschiedliche* Wirkung auf das Tailoring-Ergebnis haben würden. Spielen mehrere Einzelkriterien mit ähnlicher Wirkung auf das Tailoring-Ergebnis eine Rolle bei der projektspezifischen Anpassung, ist möglicherweise die Definition eines aggregierten Kriteriums sinnvoll.
- Kriterien zur Projektcharakterisierung sind nicht unabhängig voneinander und können sich gegenseitig beeinflussen [Har97]. Im Kriterienkatalog in Anhang A werden zu einigen Kriterien mögliche Wechselwirkungen genannt. Abhängigkeiten und gegenseitiger Ausschluss von Kriterien sollten bei der Abbildung der Kriterien auf das Konfigurationsmodell berücksichtigt werden – beispielsweise durch Cross-Tree Constraints. Zur Abbildung der Fragen zur Projektcharakterisierung auf das Konfigurationsmodell zeigen die Tabellen 5.1 und 5.2 mögliche Optionen auf.
- Die Kriterien können auf unterschiedlichen Bedeutungsebenen angesiedelt sein. Die Variabilität im Vorgehensmodell könnte etwa erlauben, die Anforderungen an das zu entwickelnde System einmal zu Beginn oder iterativ über den Verlauf des Projekts zu erheben. Ein Tailoring-Kriterium hierzu könnte *Art der Anforderungserhebung* lauten mit den Wahlmöglichkeiten *zu Beginn des Projekts* oder *iterativ*. Ein das Projekt beschreibendes Charakteristikum könnte ebenfalls *Stabilität der Anforderungen* lauten mit den Wahlmöglichkeiten *stabile Anforderungen zu erwarten* und *volatile Anforderungen zu erwarten*. Die Abbildung auf die variablen Inhalte des Vorgehensmodells würde sein, dass bei zu erwartenden stabilen Anforderungen eine Anforderungserhebung zu Beginn des Projekts gewählt würde und bei vorauszusehender Volatilität der Anforderungen eine iterative Erhebung.

In der Konsequenz bedeuten beide Kriterien dasselbe. Allerdings würde der Projektleiter sich beim ersten Kriterium aufgrund seiner eigenen Einschätzung der Situation im Projekt für eine der beiden Varianten entscheiden. Durch das zweite Kriterium würde der Projektleiter die Situation des Projekts beschreiben – die Wirkung auf das Vorgehensmodell (und damit die Einschätzung, wie sich die Charakterisierung des Projekts im Vorgehensmodell niederschlagen sollte) wäre durch den Entwickler des Vorgehensmodells bzw. des Konfigurationsmodells vorweg genommen.

Im Beispiel kann das erste Kriterium als ein das Vorgehensmodell beschreibendes aufgefasst werden, während das zweite Kriterium das Projekt beschreibt. Die Abgrenzung ist allerdings nicht immer so eindeutig möglich. Ist beispielsweise das *Kunden-/Produktvorgehensmodell* (siehe Abschnitt A.3.1) eine Eigenschaft des Projekts oder seines Vorgehensmodells? Wie verhält es sich mit der Frage, ob das zu entwickelnde System eine Benutzeroberfläche beinhaltet (siehe Abschnitt A.4)? Weiterhin kann man

6.3 Feature-basiertes Tailoring implementieren

sich bei der Charakterisierung des Projekts prinzipiell beliebig weit von direkt das Vorgehensmodell betreffenden Fragen entfernen. Im obigen Beispiel wäre etwa denkbar, von der Projekteigenschaft *Volatilität der Anforderungen* weiter zu abstrahieren und die Zusammenarbeit mit dem Kunden zu charakterisieren. Aus gewissen Eigenschaften des Kunden ließe eine hohe Volatilität der Anforderungen folgern, welche sich wiederum in einem iterativen Vorgehen zur Anforderungserhebung im Vorgehensmodell niederschlagen müsste.

Generell sollten diese Überlegungen bei der Definition der Kriterien zur Charakterisierung berücksichtigt werden, um eine Homogenität in ihrer Bedeutung zu erreichen.

Unterscheidet man rein zwischen Kriterien, die für das Projekt gelten und solchen, die für das Vorgehensmodell gelten, kann das klassische Tailoring im V-Modell XT (vergleiche Abschnitte 3.2.1 und 7.3) als das Projekt beschreibend eingeordnet werden. Die Wahl eines *Projekttypen* und einer *Projekttypvariante* haben mehrdimensionale Wirkung auf das Vorgehensmodell, wie die Wahl von *Vorgehensbausteinen* und von *Projektdurchführungsstrategien*. Auch im V-Modell XT sind die Grenzen aber fließend. Wie in Abschnitt 7.3.4 analysiert, lässt sich etwa durch das Projektmerkmal *Fertigprodukte* festlegen, ob genau ein spezielles Artefakt (*Evaluierung von Fertigprodukten*) in der projektspezifischen Ausprägung des Vorgehensmodells enthalten sein soll oder nicht. Dieses Kriterium kann als ein das Vorgehensmodell beschreibendes aufgefasst werden.

Beim Entwurf des Konfigurationsmodells sollte zu jedem Kriterium klar sein, wo und in welcher Form sich dieses auf die Inhalte des Vorgehensmodells auswirkt. Mit anderen Worten: mit welchen variablen Elementen ist das Kriterium über welche Operationen verbunden?

6.3.6 Anwendungsfall: Inhalte erstellen

Ausgehend von den Anforderungen an die Variabilität des Vorgehensmodells aus Abschnitt 6.3.3 werden die variablen Inhalte erarbeitet und im Vorgehensmodell in die geeigneten Strukturen umgesetzt. Die variablen Inhalte und deren Bezug zum Konfigurationsmodell werden vom Prozessingenieur in Zusammenarbeit mit inhaltlichen Entwicklern erarbeitet.

Akteur: Prozessingenieur, Entwickler

Auslöser: (1) Das feature-basierte Tailoring wird in einem Vorgehensmodell initial umgesetzt. (2) Bei der Anwendung des feature-basierten Tailorings hat sich der Bedarf an einer Überarbeitung oder Weiterentwicklung der variablen Inhalte im Vorgehensmodell ergeben.

Ergebnis: Die variablen Inhalte des Vorgehensmodells liegen vor und sind über die Tailoring-Operationen mit den Kriterien im Konfigurationsmodell assoziiert.

Vorbedingungen: Das Metamodell des Vorgehensmodells inklusive der Strukturen für das Konfigurationsmodell und für die variablen Inhalte liegt vor. Die Kriterien zur Charakterisierung der Projekte sind im Konfigurationsmodell beschrieben. Die Anforderungen an die umzusetzenden Inhalte sind im Umsetzungskonzept festgehalten.

Nachbedingungen: Die Anforderungen an die variablen Inhalte sind vollständig im Vorgehensmodell umgesetzt.

Die Erstellung von variablen Inhalten hängt stark ab von den konkreten Anforderungen. In jedem Fall sollte vor dem Festhalten von Vorgaben im Vorgehensmodell eine gründliche Analyse der Anforderungen der Organisation und der Bedürfnisse der Projektteams stehen. Zur Ermittlung dieser Anforderungen eignen sich die im Anwendungsfall *Bedarf analysieren* (siehe Abschnitt 6.3.3) beschriebenen Techniken.

Wie in Abschnitt 2.2.2 ausgeführt, kann das Tailoring eines Vorgehensmodells aufgefasst werden als *Anpassung in der Breite* („Wegschneiden“) und *Spezialisierung in der Tiefe* („Vertiefung“). Das feature-basierte Tailoring ermöglicht aufgrund des mächtigen Modells zur detaillierten Charakterisierung des Projekts die Aufnahme von Methoden, die nur für sehr spezifische Projektsituationen Relevanz haben.

Methodenbausteine als variable Inhalte

Da wir an dieser Stelle erstens kaum Aussagen über die Erstellung konkreter Inhalte machen können und da zweitens spezifische, situationsbezogene Vorgaben und Hilfestellungen für die Anwender eines Vorgehensmodells besonderen Mehrwert darstellen, skizzieren wir im Folgenden Möglichkeiten der Integration von Methoden(-bausteinen) und die Konsequenzen für das feature-basierte Tailoring.

In der Philosophie von EPF [Ecl11a] gibt es keine klare Trennung zwischen Vorgehensmodell und Methode. Methodenbausteine werden über eine Konfiguration zum Vorgehensmodell zusammengestellt (vergleiche Abschnitt 3.2.2). Damit stellen Methodenbausteine Variabilitätspunkte dar, die zumindest eine gewisse Verwandtschaft zu den Vorgehensbausteinen des V-Modell XT haben. Da Methoden in EPF bereits als Variabilitätspunkte aufgefasst werden können, ist auch ihre situationsabhängige Einbeziehung in die projektspezifische Ausprägung des Vorgehensmodells mit Hilfe des feature-basierten Tailorings umsetzbar.

Im V-Modell XT [RHB⁺06] ist eine „Vertiefung“ der anpassbaren Inhalte vor allem deshalb nicht möglich, weil das Konfigurationsmodell in seiner Flexibilität zu beschränkt ist, als dass sehr spezifische Projektsituationen und die damit einhergehenden Methoden-(Vorgehens-)Bausteine sinnvoll darstellbar wären (vergleiche auch Abschnitt 3.2.1).

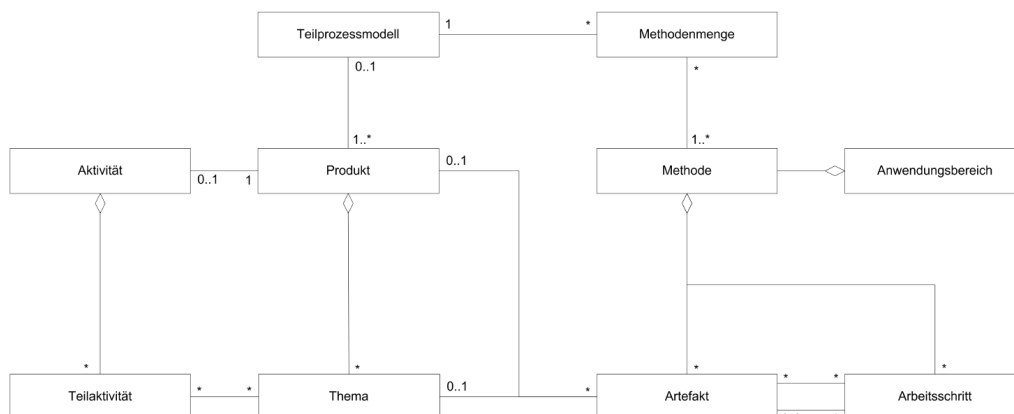


Abbildung 6.5: Metamodell zur Methodenintegration für das V-Modell XT [Ham08]

6.3 Feature-basiertes Tailoring implementieren

In [Ham08] wird ein Modell zur Integration von Vorgehensmodellen und Methodenmengen entwickelt. Am Beispiel des V-Modell XT wird gezeigt, wie die Integration in einem konkreten Vorgehensmodell durchgeführt werden kann. Im Kern besteht die Integration in einer Erweiterung des Metamodells für Vorgehensbausteine um die Strukturen zur Beschreibung von Methoden. Abbildung 6.5 stellt den entsprechenden Metamodellausschnitt nach [Ham08] dar: Methoden werden über ihre Artefakte mit den Artefakten des Vorgehensmodells in Beziehung gesetzt.

Der Ansatz in [Ham08] stellt eine Möglichkeit dar, ein *methodenneutrales* Vorgehensmodell um methodische Anteile zu erweitern. Für das Umsetzungsbeispiel V-Modell XT wird vorgeschlagen, die Auswahl von Methoden für die projektspezifische Ausprägung des Vorgehensmodells durch Definition geeigneter *Projektmerkmale* zu ermöglichen. Das erscheint unrealistisch. Projektmerkmale sind reine Ja-/Nein-Schalter, die insbesondere nicht miteinander in Beziehung stehen können. In einem realistischen Szenario wären bei der Wahl der Projektmerkmale daher viele Kompromisse zu treffen (vergleiche Abschnitt 3.2.1 und die Fallstudie zum V-Modell XT in Abschnitt 7.3 und dort insbesondere das Beispiel zur Evaluierung). Das feature-basierte Tailoring stellt hier das fehlende Glied in der Kette dar.

Da sich der Ansatz aus [Ham08] zumindest im konkreten Anwendungsbeispiel V-Modell XT nur „unterhalb“ der Variabilitätspunkte auswirkt (im Sinne des hierarchischen Aufbaus des Vorgehensmodells – es ist nur die innere Struktur eines Vorgehensbausteins betroffen), ändert der Ansatz aus Sicht des feature-basierten Tailorings nichts: Die Variabilitätspunkte bleiben die Vorgehensbausteine. Lediglich deren innerer Aufbau wird um Strukturen zur Beschreibung von Methodenanteilen erweitert.

Eine Kombination aus den Ansätzen von [Ham08] und dem feature-basierten Tailoring ermöglicht die Erstellung (und Konfiguration) spezifischer, situationsbezogener Methodeninhalte in einem Vorgehensmodell.

6.3.7 Anwendungsfall: Werkzeuge implementieren

Im Anwendungsfall *Variabilitätspunkte vereinbaren* (siehe Abschnitt 6.3.4) wird das Metamodell des Vorgehensmodells verändert. Änderungen an der Struktur des Vorgehensmodells bedingen in der Regel eine Anpassung des Werkzeugs zur Durchführung des Tailorings. Die Neuentwicklung oder Anpassung des Werkzeugs zur Durchführung des Tailorings wird von einem technischen Entwickler durchgeführt, der mit dem Metamodell des Vorgehensmodells vertraut ist.

Akteur: Technischer Entwickler

Auslöser: (1) Das feature-basierte Tailoring wird in einem Vorgehensmodell zum ersten Mal umgesetzt und macht die Entwicklung eines Werkzeugs zur Durchführung des Tailorings erforderlich. (2) Änderungen am Metamodell des Vorgehensmodells, insbesondere Änderungen am Konfigurationsmodell oder überarbeitete Variabilitätspunkte machen eine Anpassung des Werkzeugs erforderlich.

Ergebnis: Das Werkzeug zur Durchführung der projektspezifischen Anpassung basierend auf dem feature-basierten Tailoring liegt vor.

Vorbedingungen: Das Metamodell des Vorgehensmodells inklusive der Strukturen für das Konfigurationsmodell und für die variablen Inhalte liegt vor.

Die Anforderungen an das Werkzeug zum Tailoring sind im Umsetzungskonzept beschrieben.

Nachbedingungen: Das Werkzeug erfüllt die im Umsetzungskonzept festgehaltenen Anforderungen. Das Werkzeug zur Durchführung des Tailorings ist kompatibel mit dem Metamodell des Vorgehensmodells.

Die Umsetzung des Werkzeugs zur Durchführung der projektspezifischen Anpassung hängt ab von bereits vorhandenen Werkzeugen, der technischen Umgebung und den konkreten Anforderungen an das Werkzeug.

Im Folgenden sammeln wir deshalb nur einige Hinweise, die höchstwahrscheinlich für jede Umsetzung Relevanz haben.

In Abbildung 6.6 sind in Bezug auf die Benutzeroberfläche zur Durchführung des Tailorings einige Implementierungsdetails hervorgehoben, die sich in der Umsetzung des feature-basierten Tailorings für das SE Book der T-Systems International GmbH (siehe Abschnitt 7.2) als förderlich für die Benutzbarkeit des Werkzeugs erwiesen haben.

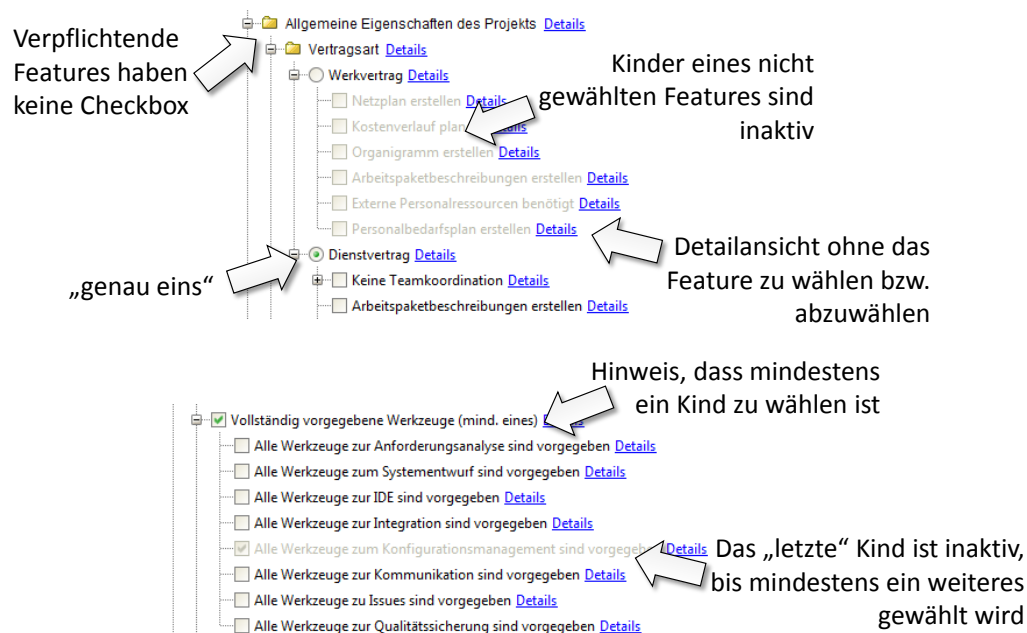


Abbildung 6.6: Beispielhafte Benutzeroberfläche zum Konfigurationsmodell

Um zu jedem Zeitpunkt eine gültige Konfiguration zu gewährleisten, sollten die in Abschnitt 5.5 beschriebenen Verfahren Anwendung im Werkzeug finden. Weiterhin hat es sich bei der Anwendung des feature-basierten Tailorings als hilfreich herausgestellt, die in Abschnitt 5.6 skizzierten Verfahren umzusetzen.

Die meisten dieser Verfahren basieren auf der Berechnung gültiger Konfigurationen mit Hilfe eines SAT Solvers. Derartige Implementierungen sollte es für jede Implementierungstechnologie geben. Stellvertretend seien hier die Java-Bibliothek Sat4J [BP10] und die Microsoft Solver Foundation [Mic12a] genannt.

Um das Konfigurationsmodell (also einen Featurebaum) in einen für einen SAT Solver geeigneten Ausdruck zu überführen, eignen sich die in Abschnitt 2.6 beschriebenen Verfahren. Ein SAT Solver kann den aussagenlogischen Ausdruck zum Konfigurationsmodell in der Regel nur auswerten, wenn sich dieser in der konjunktiven Normalform befindet. Auch die hierfür geeigneten Regeln sind in

6.4 Vorgehensmodell projektspezifisch anpassen

Abschnitt 2.6 beschrieben.

6.4 Vorgehensmodell projektspezifisch anpassen

Der Anwendungsfall *Vorgehensmodell projektspezifisch anpassen* wird vom Anwender des Vorgehensmodells – in der Regel dem Projektleiter eines beginnenden Projekts – durchgeführt. Der Projektleiter passt mit dem Anwendungsfall das Vorgehensmodell auf die Situation seines Projekts an, indem er im Werkzeug zur Durchführung des Tailorings die Charakterisierung des Projekts vornimmt und basierend darauf eine projektspezifische Ausprägung des Vorgehensmodells erstellt.

Akteur: Vorgehensmodell-Anwender

Auslöser: Ein neues Projekt soll auf der Grundlage eines Vorgehensmodells initialisiert und aufgesetzt werden. Das vorliegende Vorgehen muss in diesem Kontext auf die Projektspezifika angepasst werden.

Ergebnis: Es liegt eine projektspezifische Ausprägung des Vorgehensmodells vor. Es liegt ein Tailoring-Report vor, welcher die Entscheidungen beim Tailoring dokumentiert.

Vorbedingungen: Es gibt ein Vorgehensmodell, in welchem das feature-basierte Tailoring umgesetzt ist. Es gibt ein das Vorgehensmodell begleitendes Werkzeug zur Durchführung des projektspezifischen Tailorings.

Nachbedingungen: Das projektspezifische Vorgehensmodell ist auf die Situation des Projekts angepasst und konsistent.

Im Folgenden beschreiben wir die im Rahmen dieses Anwendungsfalls erstellten Produkte und geben Hinweise zur Durchführung.

6.4.1 Produkte

Im Anwendungsfall *Vorgehensmodell projektspezifisch anpassen* werden zwei Produkte erstellt:

Projektspezifische Ausprägung des Vorgehensmodells In der projektspezifischen Ausprägung des Vorgehensmodells sind nur diejenigen Inhalte enthalten, die aufgrund der Charakterisierung relevant für die Situation des Projekts sind.

Tailoring-Report Der Tailoring-Report fasst die Belegung der Projektcharakteristika zusammen und macht die beim Tailoring getroffenen Entscheidungen nachvollziehbar. Je nach Ausführlichkeit und Detaillierungsgrad des Konfigurationsmodells kommt der Tailoring-Report einem „Projektprofil“ nahe.

Der Tailoring-Report sollte in der Regel automatisch vom Werkzeug zur Durchführung des Tailorings generiert werden. Er ist damit ein „Nebenprodukt“ der projektspezifischen Ausprägung des Vorgehensmodells.

6.4.2 Vorgehen

Das Vorgehen zur Durchführung der projektspezifischen Anpassung hängt stark vom Werkzeug zum Tailoring ab. Ist das feature-basierte Tailoring umgesetzt,

läuft es aber in jedem Fall auf eine Wertebelegung der Projekteigenschaften im Konfigurationsmodell hinaus.

Wenn die Kriterien zur Beschreibung der Projektsituation gut gewählt sind, sollte ein Großteil der Belegung der jeweiligen Eigenschaften mit den auf das Projekt zutreffenden Werten intuitiv möglich sein. Kritisch sind diejenigen Kriterien, die eine *Einschätzung* der Situation erfordern – beispielsweise die Vertrautheit des Teams mit der eingesetzten Technologie oder die Unterstützung der Geschäftsleitung. Sieht das Konfigurationsmodell derartige Eigenschaften des Projekts vor, empfiehlt es sich für den Projektleiter unter Umständen, die Einschätzung durch Einholen mehrerer Meinungen abzusichern. Sind zur projektspezifischen Anpassung des Vorgehensmodells viele Einschätzungen vorzunehmen, könnte das Tailoring des Vorgehensmodells auch im Rahmen eines Workshops mit dem Projektteam zum Projektstart durchgeführt werden.

6.5 Beispiele

Der folgende Abschnitt enthält einige Beispiele aus der Umsetzung des feature-basierten Tailorings im SE Book. Diese Beispiele sind ein Vorgriff auf die Fallstudie zum SE Book in Abschnitt 7.2. Zusammen mit den Vorschlägen für die Umsetzung in Abschnitt 6.3 sollten die Beispiele einen Eindruck vermitteln, wie das feature-basierte Tailoring mit konkreten Projekteigenschaften und Vorgehensmodell-Inhalten aussehen kann.

Die Literaturrecherche zu Tailoring-Kriterien und deren Auswirkung auf ein Vorgehensmodell (vergleiche Abschnitt 2.4.4 und Anhang A) hat ergeben, dass existierende Vorschläge, wie ein Vorgehensmodell abhängig von der Charakterisierung des Projekts anzupassen sei, oft recht vage und abstrakt bleiben. Die Beispiele in diesem Abschnitt stellen eine konkrete Umsetzung für diese Vorschläge dar.

Alle Beispiele stammen aus der Umsetzung im SE Book und beinhalten mehr oder weniger stark Spezifika dieses Vorgehensmodells. Ziel dieses Abschnitts ist, zu verdeutlichen, wie eine Umsetzung mit Hilfe des feature-basierten Tailorings aussehen *kann*.

Beispiel 1: Konsequenzen der Vertragsart

In Abbildung 6.7 ist ein Ausschnitt aus dem Konfigurationsmodell des SE Book in der üblichen Notation für Featurebäume dargestellt. Die Art des Vertrags zwischen Auftraggeber und Auftragnehmer hat für beide Vertragspartner wesentlichen Einfluss auf das Vorgehen im Projekt (vergleiche Anhang A).

Besondere Beachtung an der in Abbildung 6.7 dargestellten Realisierung verdienen folgende Punkte:

- Abhängig von der Vertragsart kann der Projektleiter beim Tailoring einige optionale Eigenschaften des Vorgehensmodells festlegen. Einige dieser Eigenschaften, wie z.B. *Netzplan erstellen*, *Kostenverlauf planen* und *Organigramm erstellen*, können für beide Vertragsarten entschieden werden. Andere, wie z.B. *Externe Personalressourcen benötigt*, gelten nur für eine der beiden Vertragsarten.
- Im linken, unteren Bereich der Abbildung 6.7 sind einige Eigenschaften negiert formuliert. Damit wird im Konfigurationsmodell dem Umstand

6.5 Beispiele

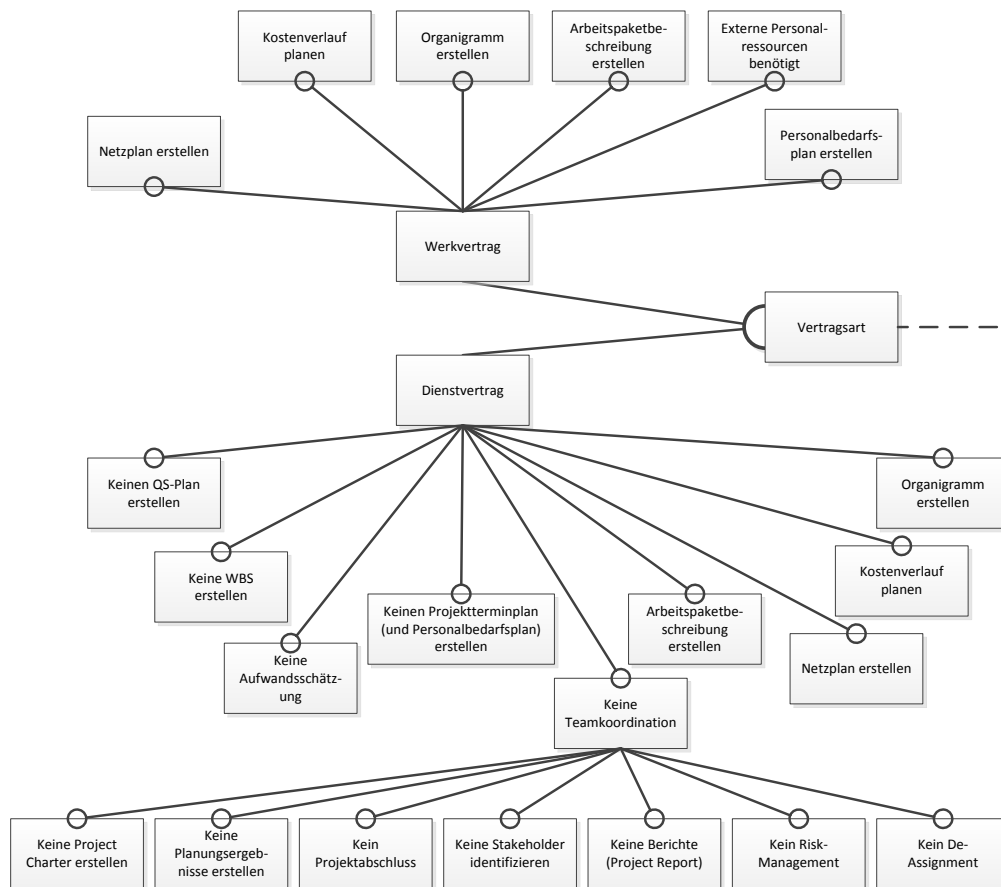


Abbildung 6.7: Vertragsart im SE Book Konfigurationsmodell

Rechnung getragen, dass *bei Wahl* der Eigenschaft Inhalte aus der projektspezifischen Ausprägung des SE Book *ausgeschlossen* werden (siehe Tabelle 6.1).

- Die Eigenschaft *Keine Teamkoordination* hat selbst keine direkten Auswirkungen auf die Inhalte der projektspezifischen Ausprägung. Sie dient rein der Gruppierung der Kind-Eigenschaften, durch welche der Projektleiter bei der projektspezifischen Anpassung individuell entscheiden kann, ob etwa ein offizieller Projektabschluss-Termin einzuplanen ist, oder ob Stakeholder im Projekt identifiziert werden müssen.

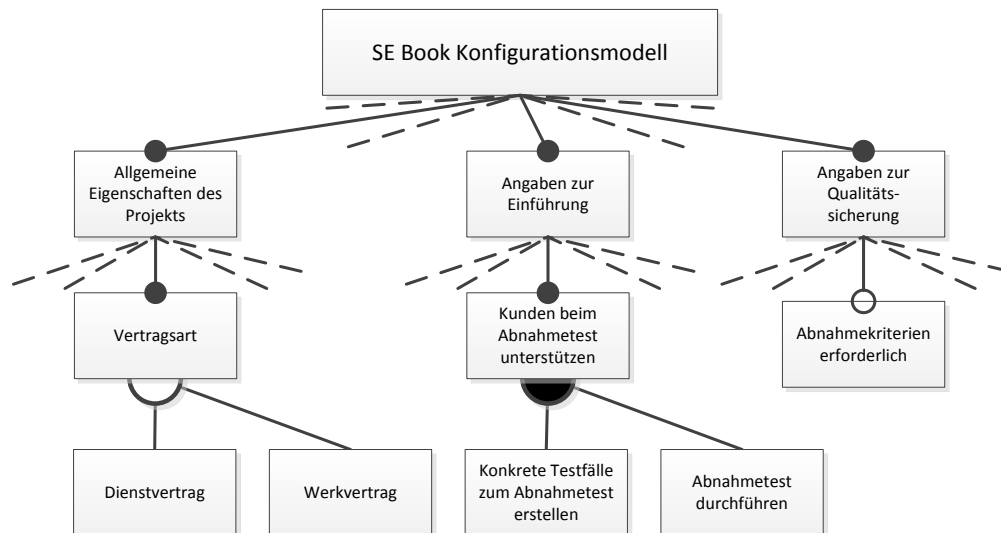
Tabelle 6.1 stellt für ausgewählte Tailoring-Kriterien aus Abbildung 6.7 die Beziehung zu den betroffenen Inhalten des SE Book dar. In der ersten Spalte befindet sich das jeweilige Tailoring-Kriterium. In der zweiten Spalte stehen die Inhalte des Vorgehensmodells, welche über die Operation *Einschließen* mit der Projekteigenschaft verbunden sind (vergleiche Abschnitt 5.4). Die dritte Spalte benennt diejenigen Inhalte, die über die Operation *Ausschließen* mit der Projekteigenschaft verknüpft sind.

Kriterium	Eingeschlossen	Ausgeschlossen
Arbeitspaketbeschreibung erstellen	<i>Ergebnis:</i> Arbeitspaketbeschreibung <i>Aktivität:</i> Arbeitspaketbeschreibung	
Keine Aufwandschätzung		<i>Ergebnis:</i> Schätzung (Aufwand) <i>Ergebnis:</i> Schätzung (Dauer) <i>Aktivität:</i> Aufwand schätzen
Kein Projektabschluss		<i>Ergebnis:</i> Projektabschlussbericht <i>Aktivität:</i> Projektabschlussbericht erstellen <i>Hilfsmittel:</i> Projektabschlussbericht
Organigramm erstellen	<i>Ergebnis:</i> Organigramm	

Tabelle 6.1: Zusammenhang zwischen Vertragsart und Vorgehensmodellinhalten

Abbildung 6.8 stellt anhand eines kleinen Ausschnitts des Konfigurationsmodells eine weitere Auswirkung der Festlegung der Vertragsart im Vorgehensmodell dar. Die Wahl der Projekteigenschaft *Dienstvertrag* führt über die unten in der Abbildung eingezeichneten Querbeziehungen zur Abwahl der Projekteigenschaften *Kunden beim Abnahmetest unterstützen* und *Abnahmekriterien erforderlich*. Damit führt umgekehrt die Wahl einer der beiden Eigenschaften zur Abwahl von *Dienstvertrag*. Da *Werkvertrag* und *Dienstvertrag* in einer *GenauEins*-Beziehung zueinander stehen, wird in diesem Fall automatisch *Werkvertrag* gewählt (vergleiche Abbildung 6.7).

6.5 Beispiele



Dienstvertrag **excludes** Kunden beim Abnahmetest unterstützen
Dienstvertrag **excludes** Abnahmekriterien erforderlichlich

Abbildung 6.8: Auswirkung der Vertragsart auf weit entfernte Bereiche des Konfigurationsmodells

Beispiel 2: Konsequenzen der Projektgröße

In der in Abschnitt 2.4.4 identifizierten Literatur zu Tailoring-Kriterien wurde das Kriterium *Team-* bzw. *Projektgröße* am häufigsten genannt. Die Empfehlung lautet üblicherweise, das Gewicht der Vorgehensweise auf die Größe des Projekts abzustimmen. Ein großes Projekt erfordert ein höheres Maß an Projektdokumentation als ein kleines (vergleiche hierzu auch Anhang A).

Auch im SE Book wird diese Empfehlung umgesetzt: durch die Wahl der Projekteigenschaft *Kleinprojekt* können einzelne Bestandteile des Vorgehensmodells, welche in kleinen Projekten nicht unbedingt erforderlich sind, ausgeschlossen werden. Ein Projekt gilt im Sinne der projektspezifischen Anpassung im SE Book als „klein“, wenn dessen Gesamtaufwand 1-2 Personenjahre nicht übersteigt. Das Kriterium *Kleinprojekt* und die damit verbundenen Einzelentscheidungen sind in Abbildung 6.9 dargestellt. Da alle Operationen auf Vorgehensmodellinhalten für diese Kriterien vom Typ *Ausschließen* sind (vergleiche Abschnitt 5.4), ist der Zusammenhang zwischen Kriterien und Vorgehensmodellinhalten für dieses Beispiel direkt in der Abbildung eingezeichnet (dargestellt durch die gestrichelten Pfeile zwischen Projekteigenschaften und Vorgehensmodellinhalten). Die Eigenschaft *Keine Dokumentation der Werkzeuganforderungen* gliedert sich in weitere Einzelentscheidungen, mit welchen für verschiedene Werkzeuge – etwa zur Anforderungsanalyse oder zur Entwicklung – angegeben werden kann, ob die Dokumentation der Anforderungen an das entsprechende Werkzeug nötig ist. Diese sind aus Platzgründen nicht in Abbildung 6.9 eingezeichnet, folgen aber dem in der Abbildung dargestellten Muster.

Neben den Entscheidungen unterhalb der Projekteigenschaft *Kleinprojekt* kann die Projektdokumentation durch eine Vielzahl weiterer Tailoring-Entscheidungen reduziert werden. Beispiele hierfür finden sich unter anderem im Ausschnitt des Konfigurationsmodells zur Vertragsart in Abbildung 6.7: insbesondere die

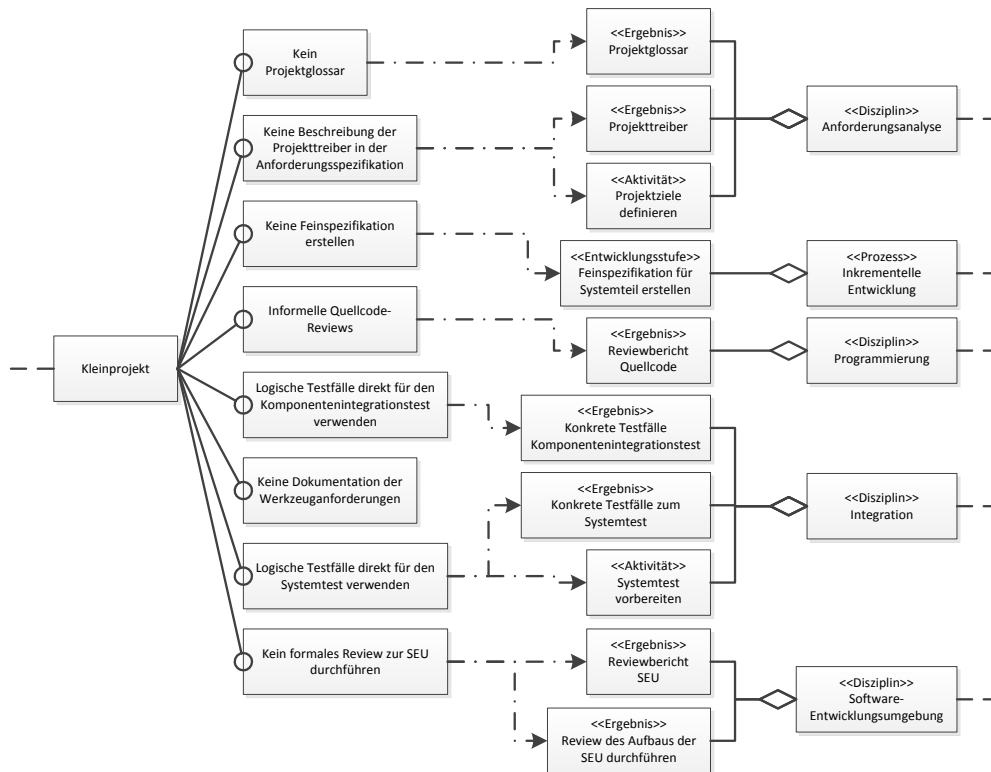


Abbildung 6.9: Zusammenhang zwischen Kleinprojekt und Vorgehensmodellinhalten

Entscheidungen unten links in der Abbildung führen alle zum Ausschluss der damit assoziierten Vorgehensmodell-Inhalte (siehe Tabelle 6.1).

Beispiel 3: Konsequenzen des Kunden-Vorgehensmodells

Die T-Systems International GmbH führt Projekte für einen Kunden durch, welcher besondere Anforderungen hinsichtlich der auszutauschenden Dokumente hat. Der Kunde erwartet Projektdokumente in einer äußeren Form und in einer Kapitelstruktur, welche von den sonst geltenden Vorgaben des SE Book abweicht. Da Projekte mit diesem Kunden häufiger durchgeführt werden, sind die besonderen Anforderungen des Kunden direkt im Vorgehensmodell verankert.

Das Tailoring-Kriterium, ob das spezielle Kunden-Vorgehensmodell im Projekt zu berücksichtigen ist, führt im SE Book zu einer Ersetzung der Vorlagen für Dokumente, die mit dem Kunden ausgetauscht werden. Es stellt damit eine Realisierung der *Replace Operation* auf dem Vorgehensmodell dar, wie sie in Abschnitt 5.3 diskutiert wurde. Tabelle 6.2 veranschaulicht den Zusammenhang zwischen dem Tailoring-Kriterium und den betroffenen Inhalten. Zur Illustration sei das Kunden-Vorgehensmodell *KV* genannt. Da alle ein- und ausgeschlossenen Inhalte mit dem Kriterium *Vorgehen nach KV* verknüpft sind ist das Kriterium nicht gesondert in Tabelle 6.2 aufgeführt.

Besondere Beachtung in Tabelle 6.2 verdienen folgende Punkte:

- Die meisten mit dem Kriterium *Vorgehen nach KV* verbundenen Operationen tauschen die Standard-Vorlage für ein Arbeitsergebnis gegen eine speziell für das Kunden-Vorgehensmodell geltende aus.

6.5 Beispiele

Eingeschlossen	Ausgeschlossen
<i>Abschnitt:</i> Tailoring für Kunden XY	
<i>Quelle:</i> Dokument 1 des Kunden	
<i>Abschnitt:</i> Ergebnisse nach KV	
<i>Abschnitt:</i> KV und SE Book	
<i>Hilfsmittel:</i> Vorlage Stufen- und Release-Plan KV	<i>Hilfsmittel:</i> Vorlage Stufen- und Release-Plan
<i>Quelle:</i> Dokument 2 des Kunden	
<i>Hilfsmittel:</i> Vorlage Anforderungsspezifikation KV	<i>Hilfsmittel:</i> Vorlage Anforderungsspezifikation
<i>Quelle:</i> Dokument 1 des Kunden	
<i>Hilfsmittel:</i> Vorlage Spezifikation KV	<i>Hilfsmittel:</i> Vorlage Spezifikation
<i>Quelle:</i> Dokument 1 des Kunden	
<i>Hilfsmittel:</i> Vorlage Architekturbeschreibung KV	<i>Hilfsmittel:</i> Vorlage Architekturbeschreibung
<i>Hilfsmittel:</i> Vorlage Software-Design KV	<i>Hilfsmittel:</i> Vorlage Software-Design
<i>Quelle:</i> Dokument 1 des Kunden	
<i>Hilfsmittel:</i> Vorlage Einführungsplan KV	<i>Hilfsmittel:</i> Vorlage Einführungsplan
<i>Hilfsmittel:</i> Vorlage Betriebshandbuch KV	<i>Hilfsmittel:</i> Vorlage Betriebshandbuch
<i>Hilfsmittel:</i> Vorlage Installationshandbuch KV	<i>Hilfsmittel:</i> Vorlage Installationshandbuch
<i>Quelle:</i> Dokument 1 des Kunden	
	<i>Hilfsmittel:</i> Vorlage High-Level Testplan
<i>Hilfsmittel:</i> Vorlage High-Level Testplan mit Testplan KV	<i>Hilfsmittel:</i> Vorlage High-Level Testplan mit Testplan
<i>Hilfsmittel:</i> Vorlage Testspezifikation KV	<i>Hilfsmittel:</i> Vorlage Testspezifikation
<i>Hilfsmittel:</i> Vorlage Testabschlussbericht KV	<i>Hilfsmittel:</i> Vorlage Testabschlussbericht
<i>Quelle:</i> Dokument 1 des Kunden	

Tabelle 6.2: Zusammenhang zwischen Kunden-Vorgehensmodell und Vorgehensmodellinhalten

- Durch das Kriterium wird auch die Dokumentation des Vorgehensmodells selbst verändert. Es werden drei neue Abschnitte in die Dokumentation aufgenommen.
- Die Quelle *Dokument 1 des Kunden* taucht in Tabelle 6.2 mehrfach auf. Der Grund für die mehrfache Einbindung wird in Abbildung 6.10 deutlich²: Die Quelle *Dokument 1 des Kunden* wird im Baum der hierarchisch organisierten Vorgehensmodell-Inhalte an unterschiedlichen Stellen referenziert.

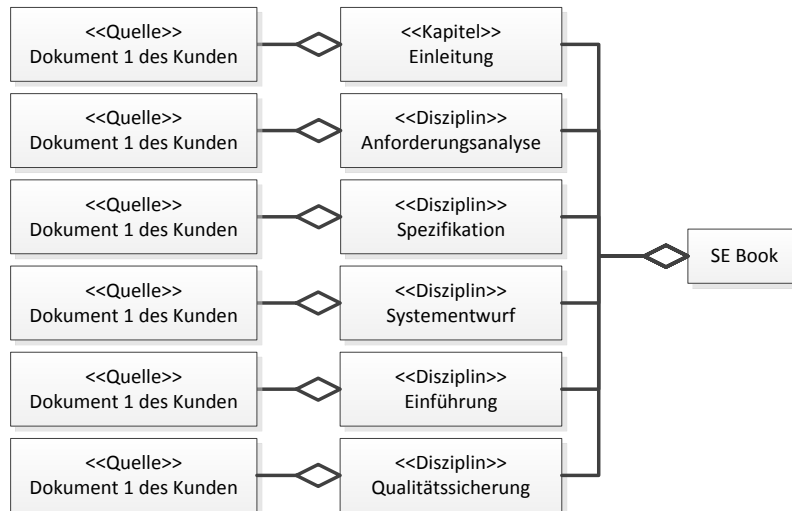


Abbildung 6.10: Dokument des Kunden wird mehrfach eingebunden

Diskussion der Beispiele

Die Beispiele in diesem Abschnitt haben *eine mögliche* Umsetzung des feature-basierten Tailorings illustriert. Wie aus den Beispielen deutlich wird, beschränkt sich die Wirkung einer Projekteigenschaft im Tailoring des SE Book vielfach auf ein einzelnes Arbeitsergebnis (und eventuell die dazugehörige Aktivität). Einerseits wird damit eine feingranulare und differenzierte projektspezifische Anpassung ermöglicht. Andererseits ist das Konfigurationsmodell im SE Book sehr umfangreich (insgesamt gibt es mehr als 100 Projekteigenschaften)³. Die in Abschnitt 5.6 vorgestellten Verfahren (siehe auch den Anwendungsfall: *Werkzeuge implementieren* in Abschnitt 6.3.7) helfen dabei, diesen Umfang für den Projektleiter bei der projektspezifischen Anpassung trotzdem beherrschbar zu machen. Die Fallstudie zum V-Modell XT in Abschnitt 7.3 ist ein Beispiel für eine Umsetzung des feature-basierten Tailorings mit einem wesentlich „gröberen“ Konfigurationsmodell. Während im SE Book nahezu alle Vorgehensmodell-Inhalte *variabel* sind, können im V-Modell XT nur Elemente vom Typ *Vorgehensbaustein* ein- oder ausgeschlossen werden.

² Das SE Book (und die eingezeichneten Disziplinen) haben natürlich mehr Inhalte als in der Abbildung dargestellt.

³ Die „Architektur“ des Konfigurationsmodells im SE Book hat zwei Gründe: Erstens muss die Rückwärtskompatibilität mit der Tailoring-Checkliste gewahrt bleiben. Mit der Tailoring-Checkliste erfolgte die Anpassung auf Basis einzelner Arbeitsergebnisse (siehe Abschnitte 3.2.4 und 7.2). Zweitens ist die Umsetzung des feature-basierten Tailorings im SE Book jung und auf die neuen Anpassungsmechanismen abgestimmte Inhalte müssen erst noch nachwachsen.

6.6 Zusammenfassung

Das vorliegende Kapitel hat die Umsetzung und Anwendung des feature-basierten Tailorings beleuchtet. Im ersten Teil des Kapitels haben wir die Anwendungsfälle zur Umsetzung des feature-basierten Tailorings detailliert beschrieben und einen beispielhaften Ablauf zu den Anwendungsfällen skizziert.

Im zweiten Teil sind wir auf die Durchführung der projektspezifischen Anpassung zu Projektbeginn eingegangen.

Im dritten Teil wurde die Umsetzung des feature-basierten Tailorings anhand konkreter Beispiele zur Verbindung der Projekteigenschaften mit den Vorgehensmodell-Inhalten plastisch veranschaulicht.

Die Umsetzung und Anwendung hängen stark vom konkreten Vorgehensmodell und den das Vorgehensmodell begleitenden Werkzeugen ab. Die Anwendungsfälle und Beispiele im vorausgegangenen Kapitel sollten aber einen Eindruck vermitteln, wie das feature-basierte Tailoring in einem Vorgehensmodell realisiert werden kann.

In diesem Kapitel führen wir einen Nachweis der praktischen Anwendbarkeit des in dieser Arbeit entwickelten Ansatzes. Wir führen hierzu zwei Studien durch, bei denen der Ansatz auf reale Vorgehensmodelle angewendet wird. Weiterhin vergleichen wir in zwei Experimenten das feature-basierte Tailoring mit alternativen Ansätzen.

In der ersten Studie wird das feature-basierte Tailoring für das SE Book der T-Systems umgesetzt.

In der zweiten Fallstudie untersuchen wir, ob und wie sich das Tailoring des V-Modell XT durch einen feature-basierten Ansatz erweitern lässt und welche Vor- und Nachteile sich daraus für das Tailoring des V-Modell XT ergeben.

Im dritten Teil dieses Kapitels führen wir ein Experiment aus Sicht der Vorgehensmodell-Anwender durch. Dabei werden die Tailoring-Ansätze des Eclipse Process Framework, des V-Modell XT und des SE Book (als Vertreter des feature-basierten Tailorings, d.h. als Ergebnis der ersten Fallstudie) gegenübergestellt. Ein ähnlich gelagertes Experiment, welches im Rahmen der Fallstudie zum SE Book durchgeführt wurde, beschreiben wir ebenfalls in diesem Teil.

Am Ende dieses Kapitels wird der Leser einen Eindruck von der Anwendbarkeit, den Vorteilen und Grenzen des in dieser Arbeit vorgestellten Ansatzes zum Tailoring von Vorgehensmodellen zur Softwareentwicklung haben.

Übersicht

7.1	Überblick	170
7.2	Fallstudie: SE Book der T-Systems	170
7.3	Fallstudie: V-Modell XT	183
7.4	Experimente zur Evaluierung	194
7.5	Zusammenfassung	204

7.1 Überblick

In diesem Kapitel beschreiben wir die Fallstudien, die wir zur Überprüfung des in dieser Arbeit entwickelten Tailoring-Ansatzes durchgeführt haben.

In Abschnitt 7.2 beschreiben wir die Erfahrungen bei der Umsetzung des feature-basierten Tailorings im hausinternen Vorgehensmodell der T-Systems – dem SE Book.

In Abschnitt 7.3 stellen wir die Ergebnisse der Anwendung des feature-basierten Tailorings auf das V-Modell XT vor.

In Abschnitt 7.4 führen wir ein Experiment zum Tailoring des Eclipse Process Framework, des V-Modell XT und des SE Book (als Vertreter des feature-basierten Tailorings) durch. Ziel dieses Experiments ist eine Bestätigung bzw. Absicherung der positiven Eigenschaften des feature-basierten Tailorings: das Experiment vergleicht die drei betrachteten Tailoring-Ansätze im Hinblick auf Transparenz bei der Durchführung des Tailorings, die Vergleichbarkeit und Nachvollziehbarkeit der Tailoring-Ergebnisse und deren Passgenauigkeit gegenüber dem Projektszenario.

Ein ähnlich aufgebautes Experiment wurde bereits im Zusammenhang mit der Fallstudie zum SE Book (vergleiche Abschnitt 7.2) durchgeführt. Aufgrund der Verwandtschaft beider Experimente beschreiben wir die Ergebnisse gemeinsam in Abschnitt 7.4.

7.2 Fallstudie: SE Book der T-Systems

Diese Studie beschreibt die Anwendung des feature-basierten Tailorings auf das *SE Book* der T-Systems International GmbH. Diese Studie wurde durchgeführt im Rahmen einer Forschungs Kooperation zwischen den Telekom Laboratories, der T-Systems International GmbH und der Technischen Universität München¹.

7.2.1 Umfeld

Die T-Systems International GmbH ist ein weltweit tätiges Unternehmen, welches eine große Bandbreite an Softwareentwicklungsprojekten durchführt. Für die Durchführung von Softwareentwicklungsprojekten verwendet die T-Systems International GmbH ein ausgereiftes, im Haus entwickeltes Vorgehensmodell – das *SE Book*. Dieses wird für alle Entwicklungsprojekte der T-Systems in Deutschland eingesetzt und für internationale Niederlassungen in einer englischsprachigen Variante ausgerollt (Stand: 2010). Weiterhin dient es als Basis für Vorgehensmodelle in weiteren Unternehmensteilen der Deutschen Telekom. Das *SE Book* hat strukturell, inhaltlich und in Bezug auf Werkzeugunterstützung einen hohen Reifegrad erreicht. Es verfügt über ein Werkzeug zur projektspezifischen Instanziierung des Vorgehensmodells, in welchem auch eine projektspezifische Anpassung anhand von vier Kriterien vorgenommen werden kann. Abbildung 3.10 stellt die Benutzeroberfläche zur projektspezifischen Anpassung des *Process Suite Generator* dar. Die Abwahl eines Kriteriums führt zum Ausschluss aller mit dem Kriterium assoziierten Vorgehensmodellinhalte aus der projektspezifischen Vorgehensmodellinstanz. Die vier Schalter bewirken also eine Ein- und Ausblendung von assoziierten Elementen.

¹ vertreten durch den Lehrstuhl Software & Systems Engineering an der Fakultät für Informatik.

Neben der Anpassung auf Basis der vier Kriterien können (und müssen) die Vorgehensmodell-Anwender ein feingranulares Tailoring auf Basis der einzelnen Arbeitsergebnisse durchführen. Dieser Tailoring-Schritt erfolgt durch Angabe zu jedem Arbeitsergebnis, ob und in welcher Form dieses im Rahmen des Projekts erstellt wird (Verwendung der Vorlage, Verwendung der inhaltlichen Struktur der Vorlage, eigene Struktur). Aus der Summe der Angaben zu jedem Arbeitsergebnis berechnet sich eine Maßzahl, die die Konformität des Projekts mit dem Vorgehensmodell widerspiegeln soll. Das feingranulare Tailoring erfolgt in einer Excel-Tabelle, welche alle Arbeitsergebnisse beinhaltet. In Abbildung 7.1 ist ein kleiner Ausschnitt der Excel-Tabelle mit Erläuterung der Spalten dargestellt.

Ist das Ergebnis verpflichtend oder optional?										
12	Werkzeuganforderungen zur Geschäftsprozessmodellierung (Modul SEU)	Optional	Werkzeuganforderungen sind bereits vorhanden. Der Abschnitt sollte erstellt werden, falls das System Geschäftsprozesse des Endkunden unterstützt und das Werkzeug zur Geschäftsprozessmodellierung nicht bereits vorhanden ist.	nein	Dokumentation erfolgt zentral im Modul SEU	nicht vorhanden	ja	nein	nein	nein
13	Werkzeuganforderungen zu KfM (Modul SEU)	Optional	Werkzeuganforderungen sollten dokumentiert werden, ausser - es handelt sich um ein Kleinprojekt oder - die Werkzeuge sind bereits vorhanden.	nein	Dokumentation erfolgt zentral im Modul SEU	nicht vorhanden	ja	nein	nein	nein
14	Werkzeuganforderungen zu Issues (Modul SEU)	Optional	Werkzeuganforderungen sollten dokumentiert werden, ausser - es handelt sich um ein Kleinprojekt oder - die Werkzeuge sind bereits vorhanden.	nein	Dokumentation erfolgt zentral im Modul SEU	nicht vorhanden	ja	nein	nein	nein
15	Anforderungsspezifikation	Mandatory		nein	Beschreibt Projektziele, Systemkontext, Randbedingungen und alle Anforderungen. Bedeutung der Inhalte ist durch [Vorkauf] genauer definiert. Die Aufkapselung kann auch Anforderungsbeschreibungen einzelner Anforderungen in jeweils separaten Produktanforderungen, auch UML, Teil der Anforderungsspezifikation	UML requirements template	ja	nein	nein	nein
16	Beschreibung der Projektziele	Mandatory	Das Ergebnis kann bei kleinen Projekten entfallen, falls - die Ziele des Projekts hinreichend genau im Scope-Statement - die Zahl der Anwender und Stakeholder überschärfbar ist und diese bereits vollständig im Stakeholder Managementplan erfasst sind	nein		UML requirements template	ja	nein	nein	nein
17										

Annotations:

- Bedeutung des Ergebnisses:** Points to the 'nein' column for rows 12-14.
- Ist das Ergebnis im Projekt relevant? Wie wird es dokumentiert?:** Points to the 'ja/nein' columns for rows 12-14.
- Inhaltlich konform, Strukturkonform, SE Book Vorlagen:** Points to the 'ja/nein' columns for rows 15-16.
- Name des Ergebnisses:** Points to the first column (row 16).
- Unter welchen Bedingungen ist das Ergebnis verpflichtend?:** Points to the third column (row 16).

Abbildung 7.1: Tailoring-Checkliste zum SE Book

Das feingranulare Tailoring auf Basis der Tailoring-Checkliste bringt einige Nachteile mit sich:

- Da das Tailoring nicht auf dem Vorgehens-Modell selbst stattfindet, kann weder strukturelle noch inhaltliche Konsistenz des Tailoring-Ergebnisses überprüft oder sichergestellt werden.
- Die Tailoring-Checkliste beinhaltet *alle* durch das SE Book vorgegebenen Arbeitsergebnisse – insbesondere auch solche, die in jedem Projekt verpflichtend zu erstellen sind. Die Liste beinhaltet also zahlreiche vermeintlich zur Disposition stehende Einträge, für die bei der Angabe der Relevanz für das Projekt eigentlich nur eine Option zulässig ist.
- Arbeitsergebnisse, die in der Tailoring-Checkliste als nicht für das Projekt relevant gekennzeichnet werden, sind in der projektspezifischen Instanz des Vorgehensmodells nach wie vor vorhanden. Das Tailoring auf Basis der Checkliste führt also nicht zu einer tatsächlichen Reduktion des Umfangs des Vorgehensmodells. Vielmehr stellt die Tailoring-Checkliste eine nachprüfbar dokumentierte Darstellung dar, *warum* im Projekt Teile der projektspezifischen Vorgehensmodellinstanz ignoriert werden.

7.2.2 Ziele

Die im SE Book vorhandenen Verfahren zur projektspezifischen Anpassung (Kriterien im Process Suite Generator und Tailoring-Checkliste) sollten in der Forschungskoooperation ausgebaut und homogenisiert werden. Insbesondere

7.2 Fallstudie: SE Book der T-Systems

war ein Ziel der Zusammenarbeit, die Anpassungsfähigkeit des Modells selbst zu steigern und robust für zukünftige Anforderungen hinsichtlich der Anpassungsfähigkeit des Vorgehensmodells zu machen.

7.2.3 Studienaufbau

Die Forschungsk Kooperation bestand aus vier Arbeitspaketen: einer Analysephase, der Entwicklung von Konzepten und technischen Mechanismen für die Anpassung des SE Book, der Entwicklung von anpassbaren Vorgehensmodell-Inhalten zur Demonstration der entwickelten Konzepte und einer Evaluierung der Ergebnisse der Kooperation.

Analysephase

In dieser Phase wurde das Verständnis für die technische Realisierung des SE Book geschaffen. Weiterhin wurden in dieser Phase alle zur Verfügung stehenden Quellen ausgewertet, um einen Eindruck über den Umgang mit dem SE Book innerhalb der T-Systems International GmbH zu erlangen. Die Details zu dieser Analyse sind in Abschnitt 2.4.3 beschrieben.

Konzepte und Mechanismen

Gegenstand dieses Arbeitspakets war die Schaffung einer technischen Basis für Tailoring des SE Book. Dabei wurden basierend auf den einleitend beschriebenen vorhandenen technischen Mechanismen folgende Ziele verfolgt:

- Zukünftige Anforderungen an die Anpassbarkeit des SE Book sollen schnell umsetzbar sein.
- Das Tailoring erfolgt werkzeuggestützt direkt auf dem zugrunde liegenden Modell.
- Abhängigkeiten zwischen Tailoring-Kriterien können dargestellt werden.
- Der Anwender wird bei der Erstellung einer inhaltlich und strukturell sinnvollen Ausprägung des Vorgehensmodells unterstützt.
- Der Prozessingenieur wird beim Entwurf des Vorgehensmodells durch Werkzeuge unterstützt und auf (möglicherweise) strukturell nicht sinnvolle Modellierung hingewiesen.

Zur Erreichung dieser Ziele wurde eine für das SE Book angepasste Variante des in der vorliegenden Arbeit erarbeiteten feature-basierten Tailorings umgesetzt. Die Anpassung bzw. Erweiterung des SE Book Metamodells für das feature-basierte Tailoring, die Implementierung des Anwenderwerkzeugs zum Tailoring im *Process Suite Generator* und ein prototypisches Werkzeug zur grafischen Bearbeitung des Konfigurationsmodells zum Tailoring wurden in diesem Arbeitspaket realisiert. Die Ergebnisse sind im folgenden Abschnitt 7.2.4 beschrieben.

Inhalte zur Demonstration

Zur Überprüfung der technischen Konzepte und der Werkzeuge zum Tailoring wurden in diesem Arbeitspaket Demonstrationsinhalte für das Vorgehensmodell erstellt. Hierzu wurde auf die Inhalte einer Best-Practice-Sammlung für

Softwareentwicklungsprojekte im Cloud-Umfeld zurückgegriffen. Diese Sammlung wurde in Kooperation zwischen den Telekom Laboratories und der Universität Duisburg-Essen erarbeitet und wurde gewählt, weil die Inhalte dieser Sammlung

- erstens bereits Bezug auf die Inhalte des SE Books nehmen und eine Integration damit wesentlich erleichtert war,
- zweitens, weil die Sammlung physikalisch als XML-Datei abgelegt ist und sich damit gut für eine maschinelle Weiterverarbeitung eignet und
- drittens, weil die Best-Practices bereits nach Projektzielen gruppiert sind.

Die im Projekt im Vordergrund stehenden Ziele eignen sich als Tailoring-Kriterien. Abhängig von der Relevanz eines Projektziels für das konkrete Projekt werden die zur Erreichung dieses Ziels empfohlenen Best-Practices in die resultierende Vorgehensmodellinstanz des SE Book eingeschlossen.

Abbildung 7.2 zeigt einen Ausschnitt der grafischen Repräsentation der Best-Practice-Sammlung für Softwareentwicklungsprojekte im Cloud-Umfeld: jeder Diamant in der linken Säule repräsentiert ein Projektziel. Die Ovale in der zweiten Säule von links repräsentieren jeweils eine Voraussetzung, die zur Erreichung der verknüpften Projektziele erfüllt sein muss. In der dritten Säule befinden sich die Best-Practices zur Erfüllung der verknüpften Voraussetzung(en) in Form von abgerundeten Rechtecken. Diese sind wiederum ganz rechts mit Inhalten des SE Book (Arbeitsschritten und Rollen) verbunden.

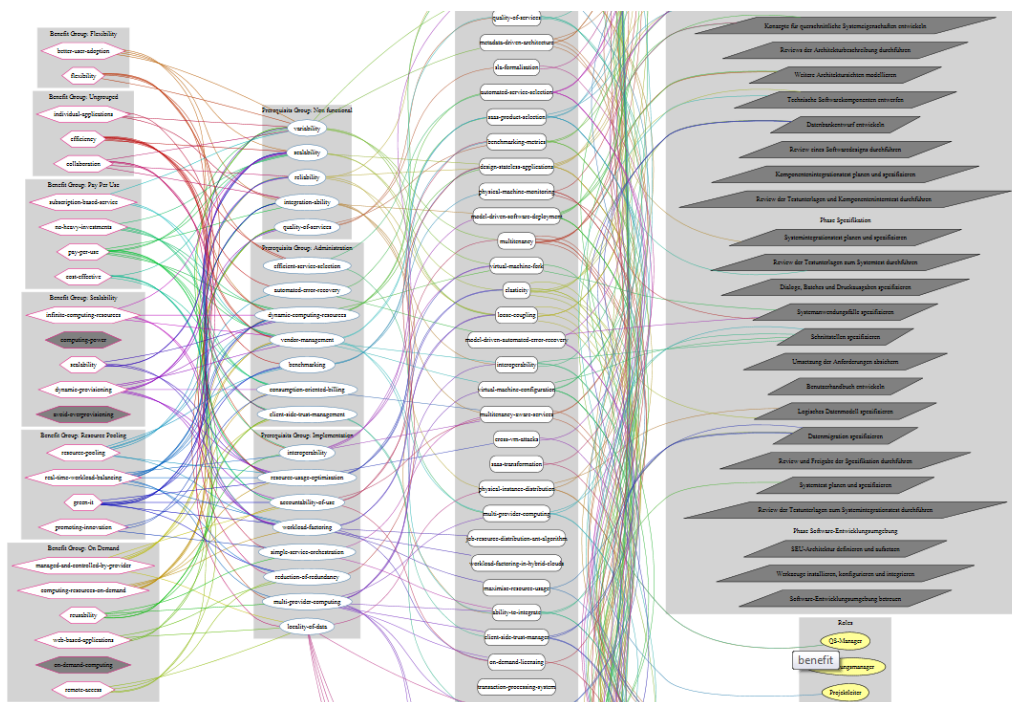


Abbildung 7.2: Ausschnitt der Best-Practices für Cloud-Projekte

Eine schematische Darstellung der Struktur der Best-Practice-Sammlung ist in Abbildung 7.3 dargestellt.

Wie die Best-Practice-Sammlung im SE Book mit feature-basiertem Tailoring umgesetzt wurde, beschreiben wir im folgenden Abschnitt 7.2.4.

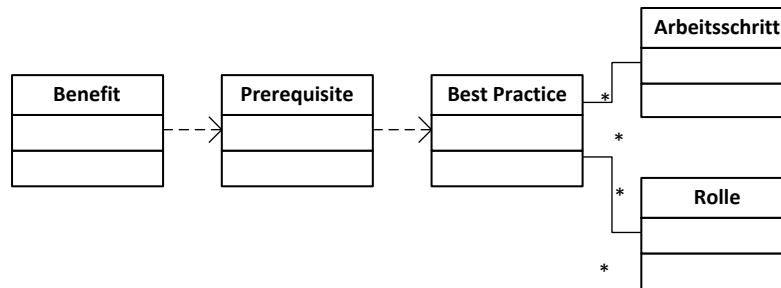


Abbildung 7.3: Struktur der Best-Practices für Cloud-Projekte

Evaluierung

Aufgrund unternehmensinterner Prozesse bei der T-Systems konnte die im Rahmen der Forschungsk Kooperation entstandene Version des SE Book und die begleitenden Werkzeuge nicht vor Ende der Kooperation im Unternehmen ausgerollt werden.

Um die Ergebnisse der Kooperation dennoch einschätzen zu können, wurde im Rahmen einer universitären Lehrveranstaltung ein Laborexperiment mit Studenten zur Verständlichkeit und Bedienbarkeit des Process Suite Generator, des EPF Composer und des V-Modell XT Projektassistenten durchgeführt. Den Aufbau und die Ergebnisse dieses Experiments beschreiben wir zusammen mit dem zweiten Experiment zur Evaluierung in Abschnitt 7.4.

7.2.4 Ergebnisse

Im Folgenden fassen wir die Ergebnisse der Forschungsk Kooperation zum Ausbau der Anpassungsfähigkeit des SE Book zusammen. Die Ergebnisse der Analysephase wurden bereits in Abschnitt 2.4.3 vorgestellt. Im Folgenden beschreiben wir deshalb nur die Ergebnisse der technischen Umsetzung und die Integration der Best-Practice-Sammlung für Softwareprojekte im Cloud-Umfeld. Die Ergebnisse des Laborexperiments werden zusammen mit den Ergebnissen der zweiten Evaluierungsstudie in Abschnitt 7.4.4 dargestellt.

Konzepte und Mechanismen

Die Ergebnisse der technischen Umsetzung des feature-basierten Tailorings im SE Book zergliedern sich in folgende Teilbereiche:

Grundlage bildet ein Metamodell des SE Books, welches die für das feature-basierte Tailoring erforderlichen Strukturen beinhaltet.

Dieses wird vom *Process Suite Generator* – dem Werkzeug für den Vorgehensmodell-Anwender – verarbeitet, um eine projektspezifisch angepasste Instanz des SE Book zu erstellen.

Wesentlicher Bestandteil einer projektspezifisch angepassten Instanz sollten angepasste Dokumentvorlagen für die Arbeitsergebnisse im Projekt sein. Da ein Großteil des Tailorings im SE Book bisher durch Ausfüllen einer Excel-Tabelle stattfand (vergleiche Abschnitt 7.2.1), war die Generierung projektspezifisch angepasster Vorlagen im Process Suite Generator zwar weitgehend realisiert aber

ungenutzt. Da im Rahmen der Forschungskoope-ration die technischen Grund-lagen für projektspezifisch angepasste Word-Vorlagen gelegt wurden, wurde dieses Thema wieder aufgegriffen und implementiert.

Das feature-basierte Tailoring bringt für den Prozessingenieur des SE Books mehr Gestaltungsspielraum gegenüber den bisherigen Möglichkeiten (siehe Abschnitt 7.2.1). Das Mehr an Möglichkeiten bedeutet aber ebenfalls die Bearbeitung eines komplexeren Modells. Um die Bearbeitung des SE Book für den Prozessingenieur dennoch handhabbar zu machen, wurde in der Forschungskoope-ration eine prototypische Variante des *Process Development Environment* (PDE) [KKWB10, KKTW10] für das SE Book umgesetzt.

SE Book Metamodell für feature-basiertes Tailoring. Im folgenden Abschnitt gehen wir auf die im Rahmen der Forschungskoope-ration gemachten Änderungen am Metamodell des SE Book ein. Das Metamodell wurde um ein Modellele-ment `FeatureConfiguration` erweitert, welches seinerseits ein Element vom Typ `Feature` enthält. Dieses Element repräsentiert die Wurzel des Konfigurationsmodells zum Tailoring des SE Book. Das Konfigurationsmodell im SE Book entspricht in seiner Grundstruktur dem in Kapitel 5 beschriebenen Konfigurationsmodell zum feature-basierten Tailoring. Lediglich in der Namensgebung der Modellelemente orientiert es sich stärker an Featurebäumen. Die wesentlichen Modellelemente rund um das `Feature` sind in Abbildung 7.4 dargestellt: ist ein Feature nicht ein Blatt im Baum, so enthält es ein Gruppierungselement, welches anzeigt, in welcher Art die Kind-Features zu kombinieren sind. Mögliche Gruppierungselemente sind `SelectOne`, `SelectAtLeastOne` und `SelectionGroup`. Über die Gruppierungselemente wird die Semantik des Featurebaums – wie in Abschnitt 2.6.1 beschrieben – realisiert. Die Gruppierungselemente beinhalten ihrerseits Elemente vom Typ `Feature`. Das Zusammenspiel der Features und der Gruppierungselemente erzeugt also die Baumstruktur.

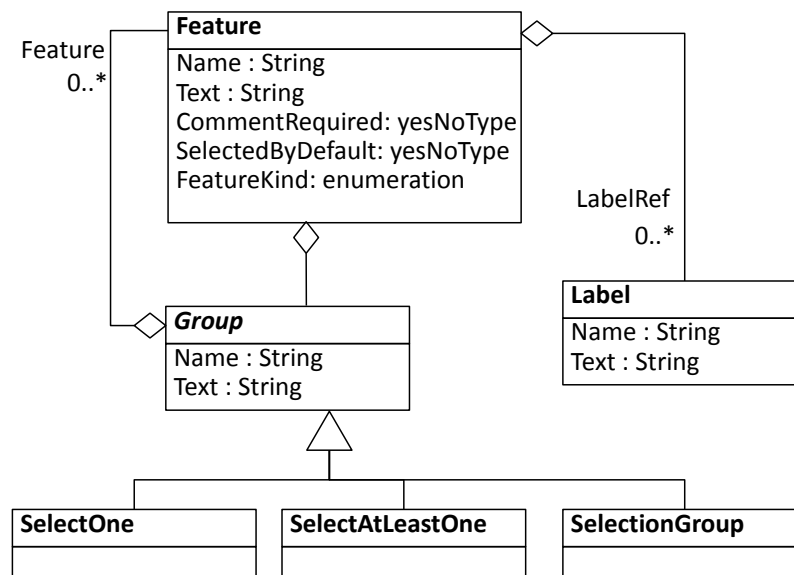


Abbildung 7.4: Feature-Baum im SE Book Metamodell

Ein Feature im SE Book Metamodell hat die folgenden Eigenschaften:

Name Der Name des Features (bzw. der Projekteigenschaft).

Text Eine ausführliche Beschreibung der Projekteigenschaft und ihrer Bedeu-

tung für das Tailoring. Dieser Text wird zu jedem Feature im *Process Suite Generator* bei der Durchführung des Tailorings angezeigt (siehe unten) und hilft dem Anwender zu entscheiden, ob die entsprechende Projekteigenschaft für das konkrete Projekt Relevanz hat oder nicht.

CommentRequired Diese Eigenschaft gibt an, ob der Anwender bei der Durchführung des Tailorings seine Wahl oder Abwahl des entsprechenden Features begründen muss.

SelectedByDefault Mit Hilfe dieser Eigenschaft kann der Prozessingenieur eine (sinnvolle) Vorbelegung der Features vornehmen. Die Vorbelegung muss nicht konsistent im Sinne des Feature-Modells sein, d.h. der Prozessingenieur kann auch sich widersprechende Vorbelegungen für einzelne Features festlegen – beispielsweise kann ein Elternfeature als nicht gewählt vorbelegt sein, seine Kinder aber schon. Der Process Suite Generator übernimmt die im Modell vereinbarte Vorbelegung für noch nicht vom Benutzer an- oder abgewählte Features, soweit dies im Sinne des Feature-Modells möglich ist (sichergestellt mit Hilfe eines SAT-Solvers – siehe unten). Würde im Beispiel der Benutzer das Elternfeature anwählen, so würde wieder die Vorbelegung ausgewertet und die Kindfeatures automatisch als Angewählt markiert. Diese Eigenschaft erspart dem Anwender beim Tailoring Arbeit, da er nur die Abweichung gegenüber der Vorbelegung konfigurieren muss.

FeatureKind Diese Eigenschaft hat nur eine Bedeutung, wenn das Feature Kind einer *SelectionGroup* Gruppe ist. Sie kann die beiden Werte *optional* und *required* annehmen und gibt an, ob das Feature ein optionales oder verpflichtendes Kind des Elternfeatures ist. Ist das Feature Kind einer *SelectOne* oder *SelectAtLeastOne* Gruppe, wird die Möglichkeit der An- und Abwahl des Features über die Semantik der Gruppe festgelegt (1 aus n im Fall von *SelectOne* und m aus n mit $m > 0$ bei *SelectAtLeastOne* – siehe auch Abschnitt 2.6.1).

Jedes Feature kann mit mehreren Elementen vom Typ *Label* assoziiert sein. Diese Elemente steuern den eigentlichen Ein- oder Ausschluss von Modellelementen in der projektspezifisch angepassten Instanz des SE Book. Vor der Beschreibung des Zusammenhangs zwischen Konfigurationsmodell und Vorgehensmodellinhalten ist in Abbildung 7.5 der Metamodellausschnitt zur Umsetzung von Querbeziehungen im Konfigurationsmodell des SE Book dargestellt.

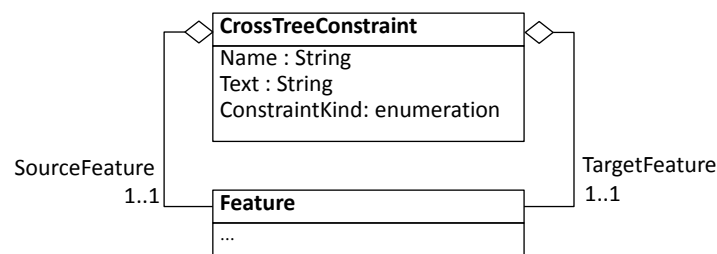


Abbildung 7.5: Cross-Tree Constraints im SE Book Metamodell

Querbeziehungen liegen im SE Book Metamodell auf Ebene der Wurzel des Konfigurationsmodells unterhalb des Elements *FeatureConfiguration*. Neben den üblichen Eigenschaften *Name* und *Text* (Beschreibungstext) gibt die Eigenschaft *ConstraintKind* an, um was für eine Art von Querbeziehung es

sich handelt. Mögliche Werte dieser Eigenschaft sind `implies` und `excludes`. In manchen Feature-Modellen kann eine Querbeziehung mehrere Quell- und Zielfeatures haben. Im SE Book Metamodell haben wir Cross-Tree Constraints so realisiert, dass sie immer nur genau ein Quell- und ein Zielfeature haben können.

Abbildung 7.6 stellt den Zusammenhang zwischen den Elementen vom Typ `Label` und den zu konfigurierenden Vorgehensmodellinhalten des SE Book dar. Der Elementtyp `AbstractProcessElementType` existiert im SE Book Metamodell nicht – er wurde zur vereinfachten Darstellung in Abbildung 7.6 eingeführt und soll alle Elementtypen repräsentieren, die am Tailoring teilnehmen können (die also die beiden Kindelemente `LabelIncludeRef` und `LabelExcludeRef` haben).

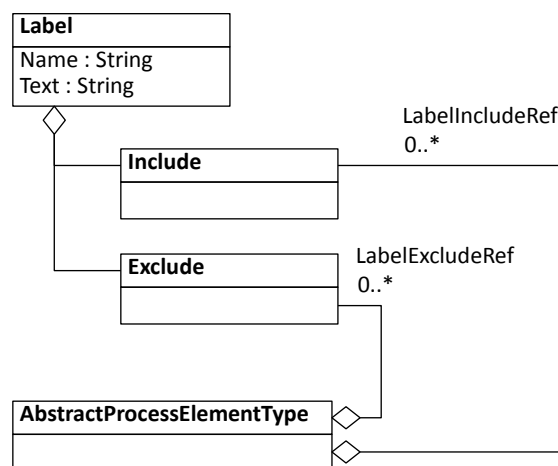


Abbildung 7.6: Zusammenhang zwischen Konfigurationsmodell und Prozessmodellelementen

Der Zusammenhang zwischen dem Konfigurationsmodell und den Elementen vom Typ `Label` wurde bereits in Abbildung 7.4 dargestellt. Labels realisieren im SE Book die in Abschnitt 5.3 beschriebenen Operationen auf den Modellinhalten: jedes Label beinhaltet zwei Kindelemente `Includes` und `Excludes`. Diese Elemente werden von variablen Inhaltselementen (in der Abbildung durch `AbstractProcessElementType` dargestellt) referenziert und tragen selbst keine weiteren Informationen.

Verweist ein variables Element auf das Include-Element eines Labels, so bedeutet dies, dass das betreffende Element bei Wahl des entsprechenden Labels (und damit bei Wahl aller mit dem Label assoziierter Features) in die resultierende Vorgehensmodellinstanz eingeschlossen wird. Wird das Label beim Tailoring abgewählt (genauer: keins der mit dem Label assoziierten Features wird ausgewählt), so wird das Vorgehensmodellelement aus der resultierenden Vorgehensmodellinstanz ausgeschlossen.

Umgekehrt verhält es sich, wenn ein Vorgehensmodellelement auf ein Exclude-Element eines Labels verweist: das Vorgehensmodellelement wird ausgeschlossen, wenn das Label ausgewählt ist und eingeschlossen, wenn das Label nicht gewählt ist.

Verweist ein Vorgehensmodellelement auf kein Include- oder Exclude-Element eines Labels, so ist es immer Teil der aus dem Tailoring resultierenden projektspezifischen Ausprägung des Vorgehensmodells.

Durch die Architektur des SE Book Metamodells ist die Information, ob und unter was für Umständen (also bei welchen Projekteigenschaften) ein Vorgehensmodellelement Teil der aus dem Tailoring resultierenden Vorgehensmodellausprägung ist, direkt beim Element selbst anzugeben. Ist keine Information angegeben, ist das Element Teil jeder möglichen Ausprägung. Das Element ist damit eine *Commonality*. Dieser Umstand erleichtert dem Prozessingenieur die Arbeit beim Entwurf des Vorgehensmodells und hält das Modell (im Vergleich zu denkbaren, alternativen Realisierungen) vergleichsweise schlank, da erfahrungsgemäß die Mehrheit der Vorgehensmodellelemente nicht vom Tailoring betroffen ist.

Werkzeuge für den Prozessanwender. Der Process Suite Generator ist ein das SE Book begleitendes Werkzeug, welches die Erstellung einer projektspezifischen Instanz des Vorgehensmodells erlaubt.

Der Anwender kann in diesem Werkzeug eine Reihe von Projekteigenschaften, wie z. B. Projektname, Geschäftsbereich, Start- und Ende-Datum, Name des Kunden, usw., angeben. Diese Eigenschaften werden in die projektspezifische Dokumentation und in die Vorlagen für Arbeitsergebnisse übernommen. Weiterhin kann der Anwender sogenannte *Vorlageneigenschaften* angeben, welche in den Metadaten der Vorlagen für Arbeitsergebnisse übernommen werden.

Ist die grobe Systemzerlegung des zu entwickelnden Systems bereits bekannt, kann der Anwender diese Systemzerlegung im *Process Suite Generator* angeben. Diese dient als Basis für eine initiale Projektplanung. Beispielsweise lässt sich angeben, dass in der ersten Projektphase die erste Version der Komponente A implementiert wird, in der zweiten Phase eine erste Version der Komponente B und beide anschließend zum Gesamtsystem integriert werden. Der resultierende Projektplan beinhaltet alle für das geplante Vorgehen vorgeschriebenen Arbeitsschritte unter Berücksichtigung der in der Systemzerlegung angegebenen Komponenten.

Schließlich erlaubt der *Process Suite Generator* das Tailoring des SE Book. Dieser Bereich des Werkzeugs wurde in der Forschungskoooperation von Grund auf neu entwickelt. Die Benutzeroberfläche des entsprechenden Reiters im Process Suite Generator ist in Abbildung 7.7 abgebildet: der hell hinterlegte Bereich in der Mitte des Fensters beinhaltet das Konfigurationsmodell zum Tailoring. Im rechten (grau hinterlegten) Bereich werden Informationen zur aktuell selektierten Projekteigenschaft angezeigt – neben einer ausführlichen Beschreibung der Projekteigenschaft werden hier insbesondere auch mögliche Seiteneffekte dargestellt (wenn die Projekteigenschaft an einer Querbeziehung teilnimmt). Im unteren Bereich auf der rechten Seite des Hauptfensters kann (oder muss) der Anwender seine Entscheidung in Bezug auf die selektierte Projekteigenschaft begründen. Ob eine Begründung der An- oder Abwahl einer Eigenschaft optional oder verpflichtend ist, kann vom Prozessingenieur zu jeder Eigenschaft individuell angegeben werden (vergleiche die Beschreibung des Metamodells oben). Wird die Eigenschaft an- oder abgewählt und hat diese Entscheidung Auswirkungen auf andere Eigenschaften, so wird der Anwender auf diese Auswirkungen zusätzlich hingewiesen. Ein derartiges Hinweisfenster ist im unteren linken Bereich der Abbildung 7.7 zu erkennen.

Um stets eine im Sinne des Konfigurationsmodells gültige Konfiguration zu gewährleisten, wurde im Process Suite Generator der SAT-Solver Sat4J [BP10] – eine Java-Implementierung von MiniSat [ES03] – verwendet. Der SAT-Solver stellt sicher, dass beim Laden des Modells eine gültige Konfiguration (basierend auf der im Modell hinterlegten Vorbelegung der Features – siehe oben) erstellt

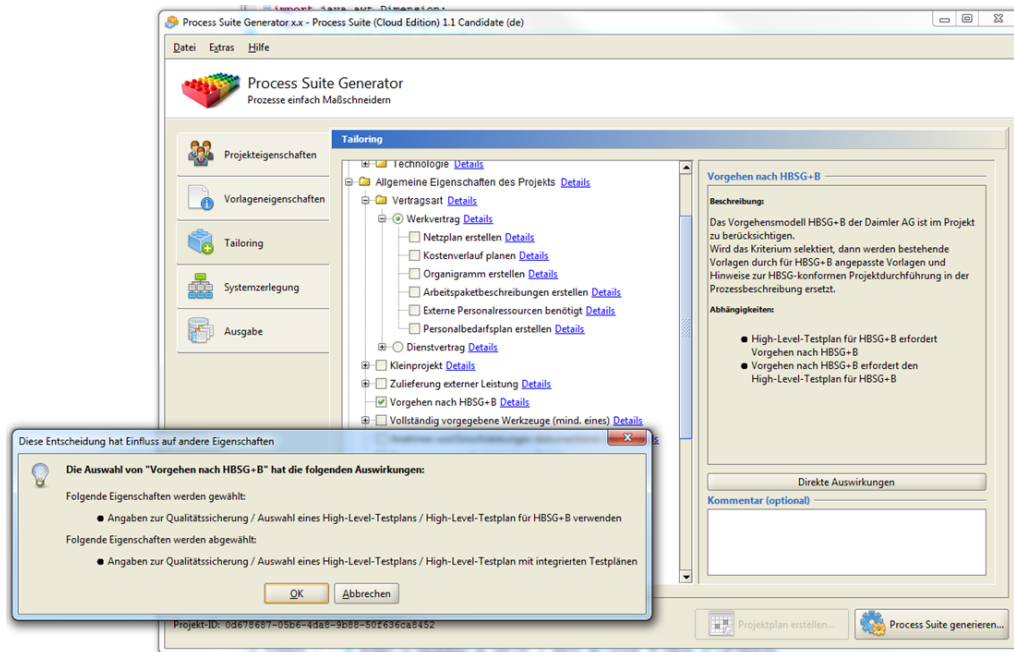


Abbildung 7.7: Überarbeitetes Tailoring im Process Suite Generator

wird. Weiterhin kommt er bei jeder Änderung der Konfiguration durch den Benutzer („bei jedem Klick“) zum Einsatz, um, ausgehend von der Wahl des Benutzers, wieder eine gültige Konfiguration herzustellen. Wählt der Anwender beispielsweise eine Eigenschaft, welche Kind einer `SelectOne` Gruppe ist, so ist die vorher gewählte Kind-Eigenschaft dieser Gruppe abzuwählen. Die Algorithmen zur Realisierung des Verhaltens im Process Suite Generator entsprechen im Wesentlichen den in den Abschnitten 5.5 und 5.6 skizzierten Verfahren.

Generierung von angepassten Word-Vorlagen. Vorlagen für Arbeitsergebnisse sind ein wesentlicher und von Anwendern eines Vorgehensmodells stark wahrgenommener Bestandteil eines Vorgehensmodells (siehe z. B. Abschnitt 2.4.3). Aus diesem Grund wurde der Process Suite Generator dahingehend erweitert, dass die Inhalte der Vorlagen das Tailoring berücksichtigen.

Die Generierung angepasster Word-Vorlagen für die Arbeitsergebnisse wurde nicht von unmittelbar an der Forschungsk Kooperation beteiligten Personen implementiert, sondern von einem Mitglied der Systems Integration Gruppe der T-Systems in Hamburg.

Werkzeuge für den Prozessingenieur. Zur Bearbeitung des SE Book wird bei der T-Systems International GmbH der *V-Modell XT Editor* verwendet. Dieser bietet einige Funktionen, die über einen reinen XML-Editor hinausgehen. Dennoch erfolgt die Bearbeitung des Modells mit diesem Werkzeug sehr nah an der physikalischen (XML-)Repräsentation des Modells. Im Rahmen der Forschungsk Kooperation haben wir deshalb eine auf das SE Book angepasste Variante des an der Technischen Universität München entwickelten Werkzeugs *Process Development Environment* (PDE) [KKWB10, KKTW10] erstellt.

Neben einer vollständigen Implementierung des SE Book Metamodells in PDE wurde hier vor allem ein grafischer Editor zur Bearbeitung des Konfigurationsmodells zum Tailoring realisiert. Ein Beispiel für die Nutzeroberfläche dieser PDE-Ausprägung findet sich in Abbildung 7.8.

7.2 Fallstudie: SE Book der T-Systems

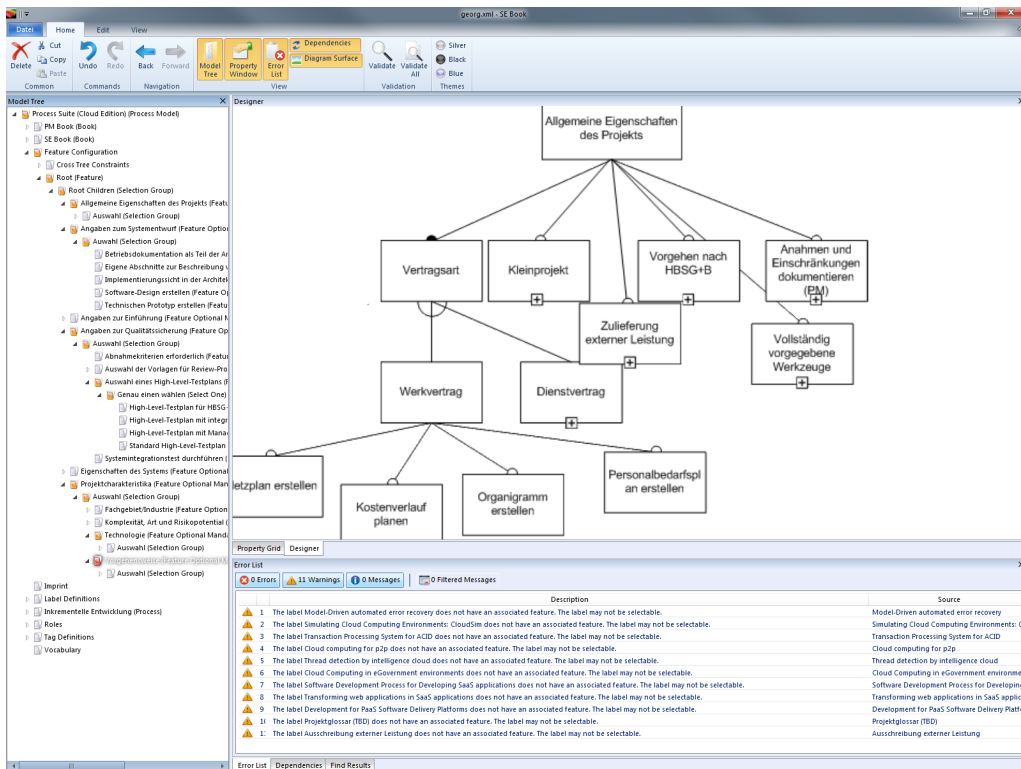


Abbildung 7.8: PDE Editor für das SE Book

Die grafische Darstellung und Bearbeitung erleichtert dem Prozessingenieur die Arbeit mit dem Modell, da Beziehungen und Abhängigkeiten zwischen Modell-elementen leichter zu erkennen sind als in der Darstellung des V-Modell XT Editor. Weiterhin ist die Bedeutung des Modells für einen mit der Notation von Feature-Modellen vertrauten Anwender sofort zu erkennen.

	Description	Source
1	The label Model-Driven automated error recovery does not have an associated feature. The label may not be selectable.	Model-Driven automated error recovery
2	The label Simulating Cloud Computing Environments: CloudSim does not have an associated feature. The label may not be selectable.	Simulating Cloud Computing Environments: C
3	The label Transaction Processing System for ACID does not have an associated feature. The label may not be selectable.	Transaction Processing System for ACID
4	The label Cloud computing for p2p does not have an associated feature. The label may not be selectable.	Cloud computing for p2p
5	The label Thread detection by intelligence cloud does not have an associated feature. The label may not be selectable.	Thread detection by intelligence cloud
6	The label Cloud Computing in eGovernment environments does not have an associated feature. The label may not be selectable.	Cloud Computing in eGovernment environme
7	The label Software Development Process for Developing SaaS applications does not have an associated feature. The label may not be selectable.	Software Development Process for Developin
8	The label Transforming web applications in SaaS applications does not have an associated feature. The label may not be selectable.	Transforming web applications in SaaS applic
9	The label Development for PaaS Software Delivery Platforms does not have an associated feature. The label may not be selectable.	Development for PaaS Software Delivery Platf
10	The label Projektglossar (TBD) does not have an associated feature. The label may not be selectable.	Projektglossar (TBD)
11	The label Ausschreibung externer Leistung does not have an associated feature. The label may not be selectable.	Ausschreibung externer Leistung

Abbildung 7.9: Hinweise für das Feature-Modell im SE Book PDE Editor

Um den Prozessingenieur auf mögliche Probleme oder Verbesserungspotenzial bei der Modellierung hinzuweisen, wurde in PDE eine Erkennung der in [vdML04] beschriebenen (möglichen) Missstände in Feature-Modellen implementiert. Die Ergebnisse dieser Arbeit sind in [CKK11] ausführlicher beschrieben. Eine beispielhafte Ausgabe der Erkennung ist in Abbildung 7.9 dargestellt.

Inhalte zur Demonstration

Um das Metamodell und den erweiterten Process Suite Generator zum Tailoring an einem praktischen Beispiel überprüfen zu können, wurde eine Best-Practice-

Sammlung für Softwareentwicklungsprojekte im Cloud-Umfeld in das erweiterte SE Book integriert.

Auch wenn sich die Best-Practice-Sammlung aus den in Abschnitt 7.2.3 beschriebenen Gründen auf den ersten Blick gut für die Integration bzw. Abbildung auf das Tailoring des SE Book eignet, stellte sich bei näherer Betrachtung heraus, dass bei der Umsetzung Kompromisse eingegangen werden müssen.

Wie Abbildung 7.2 vermuten lässt, sind die Elemente der Best-Practice-Sammlung nicht in einer Baumstruktur abgelegt. Eine *Prerequisite* etwa kann durch mehrere *Best Practices* erfüllt werden. *Best Practices* umgekehrt können zur Erreichung mehrerer *Prerequisites* dienlich sein.

Weiterhin sind die Elemente in der Best-Practice-Sammlung so stark vernetzt, dass bereits die Wahl einiger weniger übergeordneter Projektziele (*Benefits*) zur Wahl aller in der Sammlung enthaltenen Best Practices führt.

Um die Sammlung in die Baumstruktur des Konfigurationsmodells zu bringen, wurden bei der Abbildung zwei Alternativen realisiert:

Abbildung der Beziehungen über Querbeziehungen Bei dieser Variante liegen alle Elemente der Best-Practice-Sammlung flach unterhalb der Wurzel des Baums. Die Beziehungen zwischen Elementen wird über Querbeziehungen dargestellt. Abbildung 7.10 veranschaulicht diese Art der Abbildung.

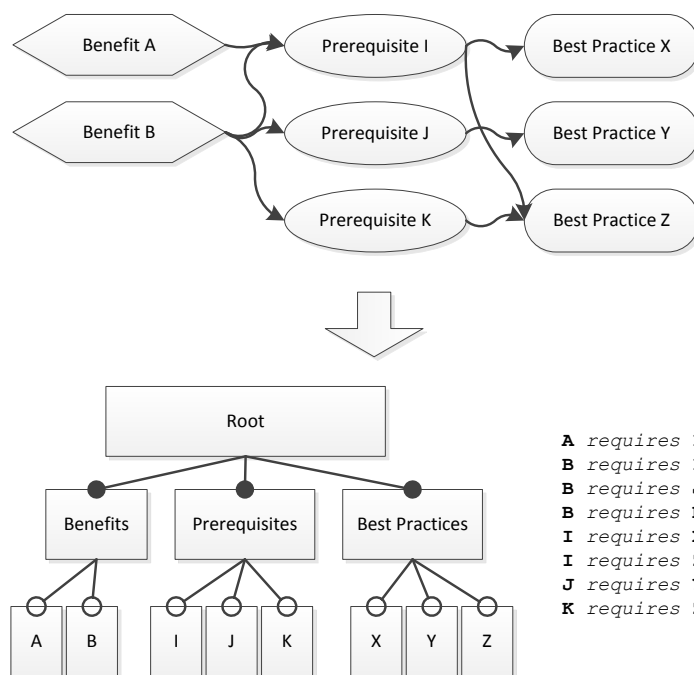


Abbildung 7.10: Abbildung der Cloud-Best-Practices mit Hilfe von Querbeziehungen

Kopie von Elementen Bei dieser Variante wird das Konfigurationsmodell aufgespannt, indem, ausgehend von den Benefits, Kopien der referenzierten Elemente angelegt werden. Abbildung 7.11 stellt diese Art der Abbildung dar.

Für die Umsetzung im SE Book haben sich die an der Forschungskoope-ration beteiligten Parteien für die zweite Variante (Kopie von Elementen) entschieden. Der Grund für diese Entscheidung liegt darin, dass die Prerequisites und die Best Practices je nach Zusammenhang – d.h. aufgrund der Wahl welchen Benefits sie Teil der projektspezifischen Instanz sind – möglicherweise unter-

7.2 Fallstudie: SE Book der T-Systems

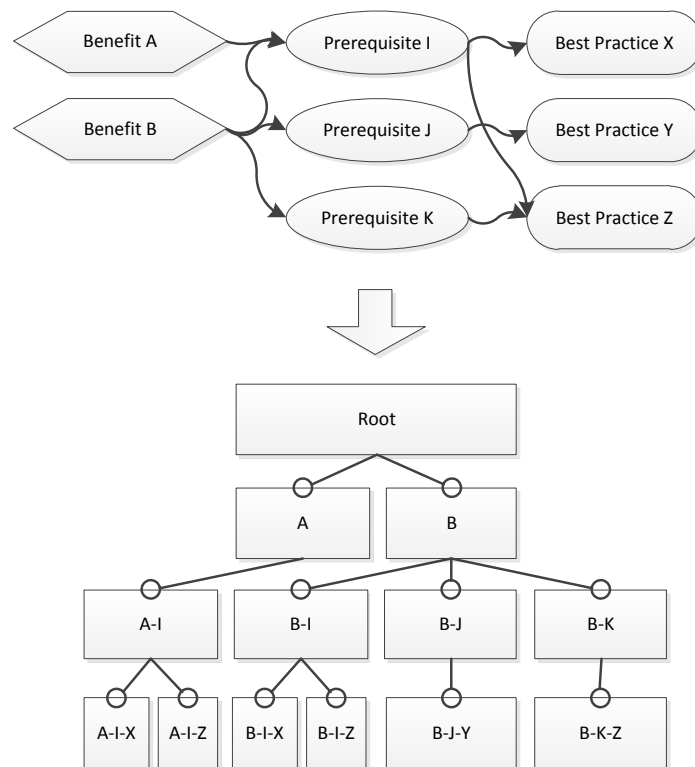


Abbildung 7.11: Abbildung der Cloud-Best-Practices durch Kopie der Elemente

schiedliche Bedeutung haben können. Wird als Benefit beispielsweise *Green IT* angestrebt, so empfiehlt die Best-Practice-Sammlung die Prerequisite *Resource Usage Optimization*. Diese wird ebenfalls empfohlen, wenn der Benefit *Real-time Workload Balancing* ein Ziel ist. *Resource Usage Optimization* hat damit je nach übergeordnetem Ziel eine unterschiedliche Qualität, die sich beim Tailoring potenziell in unterschiedlicher Weise niederschlägt (bzw. niederschlagen sollte).

7.2.5 Diskussion

Das SE Book der T-Systems war vor der in der Studie beschriebenen Forschungs-kooperation ein ausgereiftes Vorgehensmodell, welches unternehmensweit eingesetzt wird. Die Erstellung einer projektspezifischen Instanz wird für Anwender des SE Book durch das Werkzeug *Process Suite Generator* wesentlich vereinfacht. Die Möglichkeit zur projektspezifischen Anpassung war allerdings nur durch die Wahl vier einfacher Kriterien im *Process Suite Generator* durchzuführen. Ein Großteil der projektspezifischen Anpassung musste durch Ausfüllen einer Tailoring-Checkliste in Form einer Excel-Tabelle – ohne direkten Bezug zum zugrunde liegenden Modell – durchgeführt werden. Im Zuge der Forschungs-kooperation haben wir die Anpassungsfähigkeit des SE Book in das Modell verlagert und durch die Verwendung des in der vorliegenden Arbeit entwickelten Ansatzes zum feature-basierten Tailoring mit klaren Regeln ausgestattet. Durch Maßnahmen, wie beispielsweise der Vorbelegung von Projekteigenschaften, bleibt auch das potenziell umfangreichere Tailoring für die Anwender im *Process Suite Generator* handhabbar. Rückmeldungen von Pilotprojekten innerhalb der T-Systems und das im Rahmen einer universitären Lehrveranstaltung durchgeführte Laborexperiment (siehe Abschnitt 7.4) lassen den Schluss

7.3 Fallstudie: V-Modell XT

sein können. Sollte beispielsweise ein einzelnes Produkt (mit den begleitenden Elementen, wie der bearbeitenden Aktivität) Gegenstand des Tailorings sein, so müsste man für dieses Produkt einen Vorgehensbaustein anlegen, welcher nur dieses Produkt enthält.

7.3.2 Ziele

In dieser Studie sollte das Metamodell des V-Modell XT so verändert werden, dass es anstatt der Projekttypen, Projekttypvarianten und Projektmerkmale ein Feature-Modell zum Tailoring verwendet. Ziel war dabei neben einem Nachweis der Machbarkeit eine Analyse und Bewertung im Vergleich zur bisherigen Situation. Lassen sich die bisherigen Strukturen auf ein Feature-Modell zum Tailoring abbilden? Bringt das feature-basierte gegenüber dem bisherigen Tailoring im V-Modell XT einen Mehrwert?

7.3.3 Studienaufbau

Diese Studie wurde weitgehend im Rahmen einer studentischen Arbeit [Stö11] durchgeführt. Betrachtungsgegenstand war das V-Modell XT in der Version 1.3. Die Studie gliederte sich in folgende Arbeitspakete:

1. Analyse der Ist-Situation. Wie ist das Tailoring im V-Modell XT 1.3 derzeit realisiert?
2. Umsetzung des feature-basierten Tailorings im V-Modell XT Metamodell 1.3.
3. Abbildung des V-Modell XT Standard auf die neuen Metamodellstrukturen.
4. Untersuchung, inwieweit die Variabilitätspunkte beim Tailoring des V-Modell XT feingranularer gewählt werden können. Kann man als Variabilitätspunkte anstatt der Vorgehensbausteine, Ablaufbausteine und Ablaufbausteinspezifikationen z.B. einzelne Produkte zulassen? Welche Konsequenzen hätte dies?
5. Diskussion und Gegenüberstellung des „alten“ und des „neuen“ Tailorings anhand eines fiktiven, illustrativen Szenarios.

Aufgrund der Verwandtschaft der technischen Basis des SE Book und des V-Modell XT wurde bei der Umsetzung des feature-basierten Tailorings im Metamodell des V-Modell XT auf die bestehenden Ergebnisse der Fallstudie zum SE Book (Abschnitt 7.2) zurückgegriffen.

7.3.4 Ergebnisse

Im Folgenden fassen wir die Ergebnisse der Fallstudie zum feature-basierten Tailoring des V-Modell XT zusammen. Die Ergebnisse lassen sich grob in folgende Teilbereiche einteilen:

- Integration eines Konfigurationsmodells zum Tailoring in das V-Modell XT und die dazu notwendigen Änderungen am Metamodell.
- Abbildung und Migration des V-Modell XT Standard auf das geänderte Metamodell.
- Evaluierung der Ergebnisse anhand eines fiktiven Beispiels.

7.3 Fallstudie: V-Modell XT

sondere der Vorgehensbausteine – um beim Tailoring präziser Einfluss auf spezifische Elemente des Vorgehensmodells nehmen zu können.

In der Studie sollte festgestellt werden, ob feingranulare Modellelemente anstatt der Vorgehensbausteine – beispielsweise *Produkte* – im V-Modell XT als Variabilitätspunkte in Frage kommen. Da einem Produkt im V-Modell XT immer eine das Produkt bearbeitende Aktivität zur Seite steht, kommt die Entfernung eines einzelnen Produktelements aus einer projektspezifischen Vorgehensmodellinstanz nicht in Frage: der Variabilitätspunkt muss so gewählt sein, dass ein Produkt und seine begleitenden Elemente – neben der bearbeitenden Aktivität z.B. noch die Themen des Produkts und Beziehungen des Produkts zu anderen Modellelementen – zusammen einer Tailoring-Operation unterworfen sind. Ruft man sich die Inhalte eines Vorgehensbausteins aus Abbildung 4.9 in Erinnerung, stellt man fest, dass ein Vorgehensbaustein exakt das Modellelement ist, welches Produkte und die begleitenden Elemente sinnvoll in einer Einheit kapselt. Sollen beim Tailoring die Konventionen des V-Modell XT gewahrt bleiben (beispielsweise „zu jedem Produkt gibt es eine bearbeitende Aktivität“ und „zu jeder Aktivität gibt es genau ein Produkt“), so ist ein Vorgehensbaustein die kleinste variable Einheit und damit der einzig sinnvolle Variabilitätspunkt.

Ein feingranulares Tailoring auf Ebene einzelner Produkte ist möglich, indem ein Vorgehensbaustein erstellt wird, welcher nur das variable Produkt enthält. Betrachtet man die Anzahl der Produkte in den Vorgehensbausteinen des V-Modell XT Standard in Tabelle 7.1, findet genau diese Praxis beim Tailoring des V-Modell XT bereits Anwendung: der Vorgehensbaustein *Evaluierung von Fertigprodukten* enthält ein einziges Produkt *Marktsichtung für Fertigprodukte*, welches nur in der projektspezifischen Instanz des V-Modell XT enthalten ist, wenn das Projektmerkmal *Fertigprodukte* den Wert *Ja* hat. Der Vorgehensbaustein *Sicherheit* beinhaltet selbst sogar überhaupt keine Produkte. Er wird beim Tailoring berücksichtigt, wenn das Projektmerkmal *Systemsicherheit* den Wert *Ja* hat und führt dazu, dass Produkte aus anderen Vorgehensbausteinen um Themen zur Safety und Security erweitert werden.

Als Ergebnis unserer Untersuchungen bleibt festzuhalten, dass Vorgehensbausteine im V-Modell XT das geeignete Modellelement für Variabilität der statischen Modellinhalte sind. Das Metamodell des V-Modell XT blieb im Rahmen unserer Studie in dieser Hinsicht unverändert.

Abbildung des V-Modell XT Standards. Wesentliches Ziel der Studie war die Feststellung, ob sich die Strukturen des klassischen Tailorings im V-Modell XT auf das Konfigurationsmodell des feature-basierten Tailorings abbilden lassen. *Projekttypen* und *Projekttypvarianten* bilden in ihrer Struktur bereits einen Baum. Spielraum gibt es bei der Abbildung bei der Darstellung der *Projektmerkmale* im Konfigurationsmodell: diese können im V-Modell XT sowohl von Projekttypen als auch von Projekttypvarianten „angezogen“ werden. Für die Abbildung der Tailoring-Strukturen des V-Modell XT auf ein Konfigurationsmodell eröffnen sich daher zwei Alternativen: die Projektmerkmale werden von Projekttypen oder Projekttypvarianten über eine bedingende Querbeziehung referenziert oder jedes Projektmerkmal wird für die Projekttypen und Projekttypvarianten, für die es relevant ist, kopiert.

Tabelle 7.2 fasst die Abbildungsregeln für die Abbildung mit Hilfe von Querbeziehungen zusammen. In der Tabelle werden folgende Abkürzungen verwendet: PT: Projekttyp, PTV: Projekttypvariante, PM: Projektmerkmal, PMW: Projektmerkmalswert, $PT \Rightarrow PM$: das Projektmerkmal wird durch einen Projekttypen „angezogen“ und $PTV \Rightarrow PM$: das Projektmerkmal wird durch eine Projekt-

Vorgehensbaustein	Anzahl Produkte
Projektmanagement	11
Qualitätssicherung	7
Konfigurationsmanagement	4
Problem- und Änderungsmanagement	4
Lieferung und Abnahme (AG)	3
Vertragsschluss (AG)	10
Anforderungsfestlegung	3
Evaluierung von Fertigprodukten	1
Sicherheit	0
Sicherheit (AN)	3
Kaufmännisches Projektmanagement	2
Messung und Analyse	2
Lieferung und Abnahme (AN)	1
Vertragsschluss (AN)	6
Systemerstellung	18
HW-Entwicklung	8
SW-Entwicklung	9
Logistikkonzeption	6
Benutzbarkeit und Ergonomie	4
Weiterentwicklung und Migration von Altsystemen	2
Einführung und Pflege eines organisationsspezifischen Vorgehensmodells	4
Multi-Projektmanagement	2

Tabelle 7.1: Anzahl der Produkte pro Vorgehensbaustein

7.3 Fallstudie: V-Modell XT

typvariante „angezogen“. „in XOR-Gruppe“ besagt, dass von den unmittelbaren Geschwistern der Projekteigenschaft *genau ein* zu wählen ist. Im Metamodell entspricht das der `GenauEins` Gruppierung.

Element im klassischen V-Modell XT Tailoring	Repräsentation im Konfigurationsmodell
PT	Kind unterhalb der Wurzel; in XOR-Gruppe.
PTV	Kind unterhalb eines PT; in XOR-Gruppe.
PM	Optionales Kind unterhalb der Wurzel.
PMW	Kind unterhalb eines PM; in XOR-Gruppe.
PT \Rightarrow PM	Bedingende Querbeziehung zwischen PT und PM
PTV \Rightarrow PM	Bedingende Querbeziehung zwischen PTV und PM

Tabelle 7.2: Abbildung des klassischen V-Modell XT Tailorings mit Querbeziehungen

In dem im V-Modell XT verwendeten Konfigurationsmodell erlauben wir aus implementierungstechnischen Gründen nur jeweils eine „Gruppierungsart“ von Kind-Projekteigenschaften. So erlaubt das Konfigurationsmodell etwa nicht, unterhalb einer Eltern-Eigenschaft die Kombination von sich gegenseitig ausschließenden und optionalen Kind-Eigenschaften. Aus diesem Grund müssen zur Realisierung der Abbildungsvorschrift aus Tabelle 7.2 zwei künstliche Eigenschaften `Projekttypen` und `Projektmerkmale` eingeführt werden. Die Kinder unterhalb von `Projekttypen` stehen untereinander in einer XOR-Beziehung; die Kinder der Eigenschaft `Projektmerkmale` sind alle optional (im Metamodell durch eine `Auswahl` Gruppierung dargestellt).

Ein beispielhaftes, aus der Abbildungsvorschrift in Tabelle 7.2 hervorgehendes Konfigurationsmodell findet sich in Abbildung 7.14.

In Tabelle 7.3 ist die Abbildungsvorschrift bei Kopie der Projektmerkmale für jeden der referenzierenden Projekttypen und jede der referenzierenden Projekttypvarianten dargestellt. Die verwendeten Abkürzungen entsprechen denen aus Tabelle 7.2.

Abbildung 7.15 stellt ein beispielhaftes Konfigurationsmodell dar, welches der Abbildungsvorschrift aus Tabelle 7.3 folgt.

Auch bei dieser Art der Abbildung sind aufgrund der Einschränkung, dass Kinder einer Projekteigenschaft nur auf eine Art und Weise kombiniert werden dürfen, künstliche Eigenschaften einzuführen: die Projekttypvarianten unterhalb eines Projekttyps stehen in einer XOR-Beziehung, während die unterhalb eines Projekttypen aufgehängten Projektmerkmale optional sind. Um die unterschiedliche Art der Kombination der Kinder unterhalb eines Projekttypen abzubilden, sind deshalb die in Abbildung 7.15 unterhalb des Projekttypen `PT B` dargestellten Eigenschaften `Projekttypvarianten` und `Projektmerkmale` einzuführen.

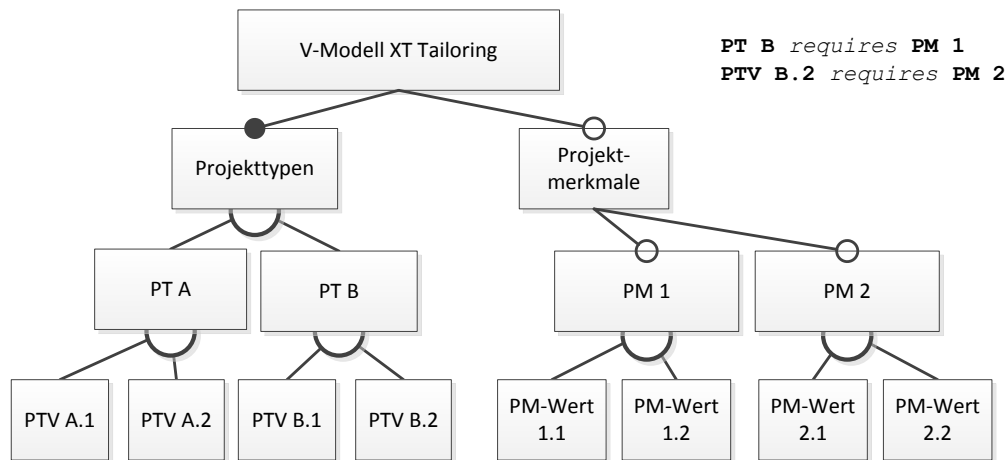


Abbildung 7.14: Beispiel für die Abbildung des klassischen V-Modell XT Tailorings mit Querbeziehungen

Element im klassischen V-Modell XT Tailoring	Repräsentation im Konfigurationsmodell
PT	Kind unterhalb der Wurzel; in XOR-Gruppe.
PTV	Kind unterhalb eines PT; in XOR-Gruppe.
PM	optionales Kind unterhalb eines PT oder einer PTV.
PMW	Kind unterhalb eines PM; in XOR-Gruppe.
PT \Rightarrow PM	Kopie des PM wird als optionales Kind unterhalb des PT angelegt.
PTV \Rightarrow PM	Kopie des PM wird als optionales Kind unterhalb der PTV angelegt.

Tabelle 7.3: Abbildung des klassischen V-Modell XT Tailorings mit Kopien der Projektmerkmale

7.3 Fallstudie: V-Modell XT

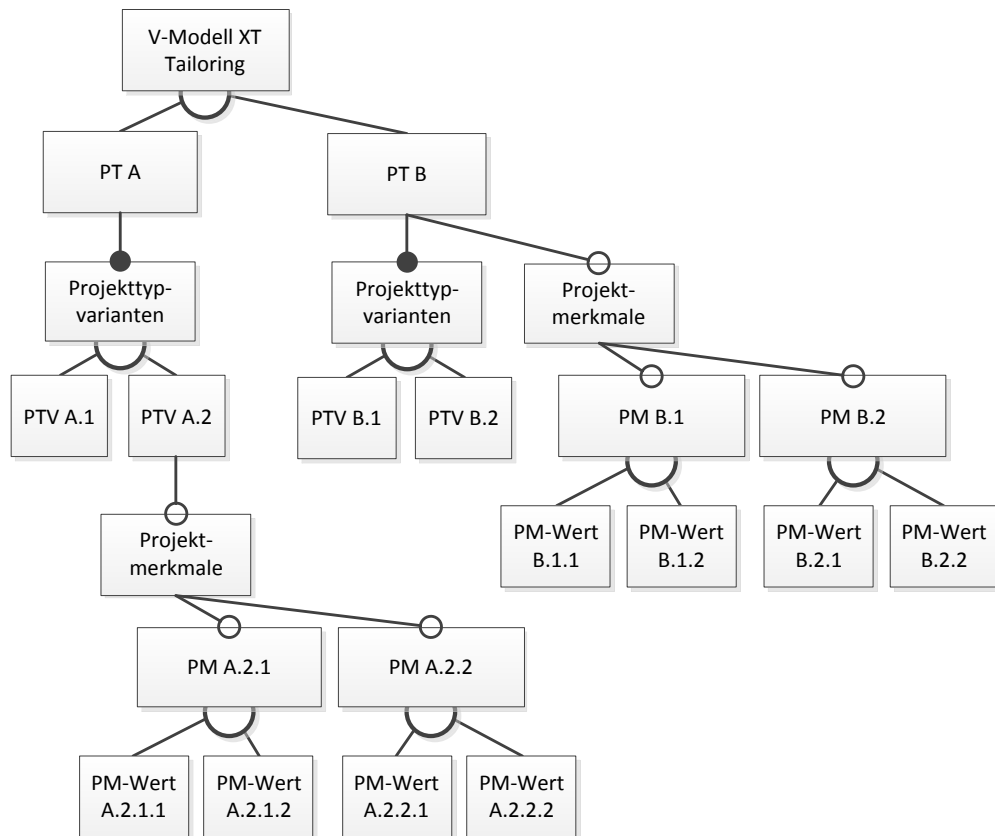


Abbildung 7.15: Beispiel für die Abbildung des klassischen V-Modell XT Tailorings mit Kopien der Projektmerkmale

Beide Abbildungsvarianten wurden im Rahmen der Studie in einem Migrationswerkzeug umgesetzt, welches die Inhalte eines bestehenden V-Modell XT vom alten auf das neue Metamodell migriert. Das Ergebnis der Migration für die Variante der Kopie von Projektmerkmalen ist in Abbildung 7.16 dargestellt. Zu erkennen ist dabei beim Migrationsergebnis im rechten Teil der Abbildung insbesondere, dass in der XML-Repräsentation alle Strukturen zum Tailoring unterhalb eines Knotens auf oberster Ebene (Projekteigenschaften) zusammengefasst sind und sich nicht mehr über mehrere Knoten auf oberster Ebene verstreuen.

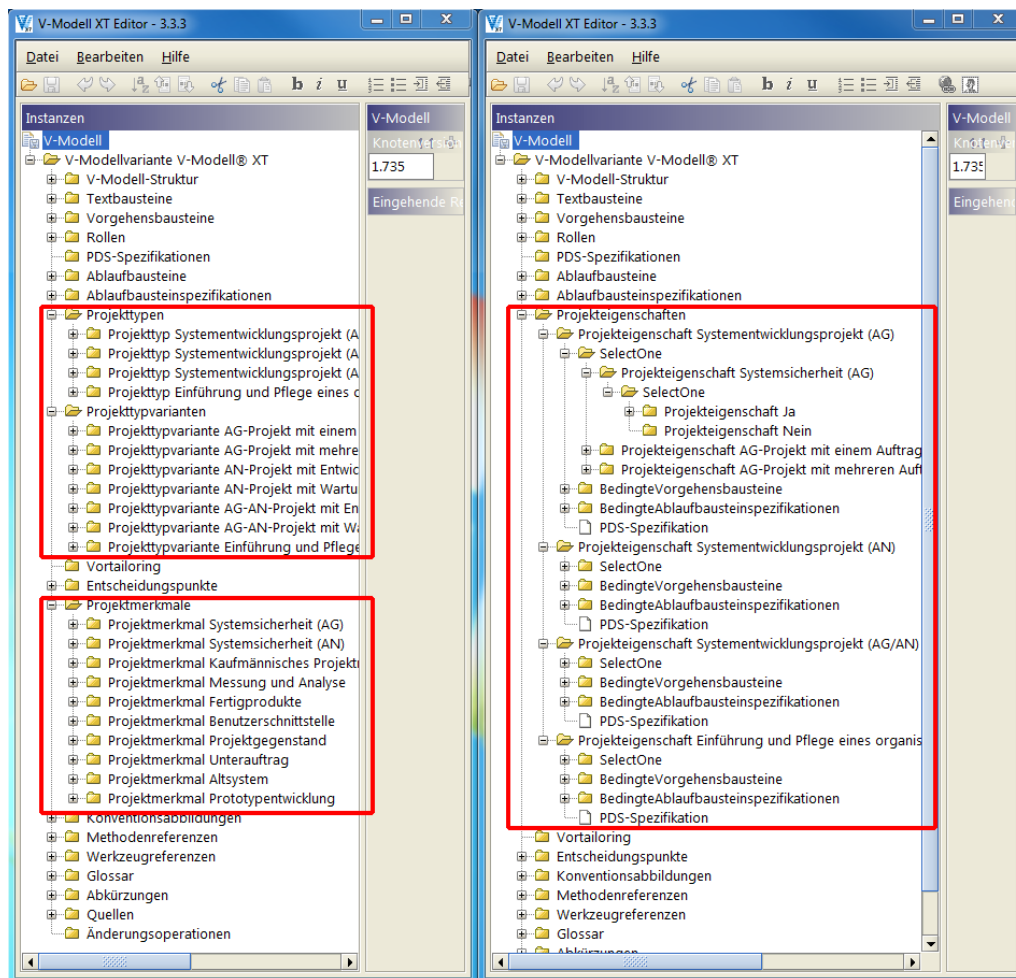


Abbildung 7.16: Migration des V-Modells auf Metamodell zum feature-basierten Tailoring [Stö11]

Da die Variante der Kopie von Projektmerkmalen das Konfigurationsmodell sehr schnell aufbläht, ist in der Praxis die Variante der Referenzierung von Projektmerkmalen über Querbeziehungen zu bevorzugen. Zwar könnte man sich vorstellen, dass ein gleich lautendes Projektmerkmal je nach Kontext (also z.B. je nach gewähltem Projekttyp) eine unterschiedliche Bedeutung hat – im V-Modell XT hat ein Projektmerkmal aber unabhängig vom Kontext immer die selben Auswirkungen auf die resultierende Vorgehensmodellinstanz (also die gewählten Vorgehensbausteine, Ablaufbausteine und Ablaufbausteinspezifikationen). Aus diesem Grund ist eine Kopie von Projektmerkmalen für die unterschiedlichen Teilbäume des Konfigurationsmodells nicht erforderlich – im Gegenteil:

7.3 Fallstudie: V-Modell XT

mehrere Projekteigenschaften im Konfigurationsmodell, die ein und dieselbe Bedeutung und Auswirkung auf die resultierende Vorgehensmodellinstanz haben, erschweren die Wartbarkeit und Übersichtlichkeit des Konfigurationsmodell.

Fiktives Beispiel zur Evaluierung. Zur Diskussion und Gegenüberstellung des klassischen V-Modell XT Tailorings und des feature-basierten Ansatzes wurde im Rahmen der Studie ein einfaches fiktives Szenario entwickelt. In dem Szenario wurde angenommen, dass eine organisationsspezifische Variante des V-Modell XT für einen IT-Dienstleister zu erstellen sei. Dieser träte in erster Linie als Auftragnehmer auf, nimmt aber gegenüber Unterauftragnehmern auch die Rolle des Auftraggebers ein. Das Vorgehensmodell soll von den Projektleitern an ihr jeweiliges Projekt anpassbar sein. Als übergeordnetes Tailoring-Kriterium wird also die Wahl zwischen Auftraggeber- und Auftragnehmerprojekt festgestellt.

Für den Typ Auftraggeberprojekt soll das Projekt durch die in Tabelle 7.4 dargestellten Eigenschaften charakterisiert sein.

Eigenschaft	Typ	Mögliche Werte
Ansprechpartner ist bei Auftragnehmer vor Ort	optional	Ja / Nein
Ansprechpartner führt vor Ort regelmäßige Zwischenabnahmen durch	verpflichtend	Ja / Nein

Tabelle 7.4: Fiktive Projekteigenschaften bei Auftraggeberprojekt für IT-Dienstleister

Die Angabe, ob ein Ansprechpartner beim Auftragnehmer vor Ort zur Verfügung gestellt wird, ließe sich im Tailoring des V-Modell XT nur über eine Projekteigenschaft abbilden, da diese Angabe optional sein soll. Wenn allerdings ein Ansprechpartner abgestellt wird, soll beim Tailoring angegeben werden, ob dieser regelmäßige Abnahmen durchführt. Im Vorgehensmodell hätte dies beispielsweise Auswirkungen auf Art und Umfang der vom Auftragnehmer zu erbringenden Projektdokumentation. Im Tailoring des V-Modell XT bietet sich für diese Eigenschaft die Erstellung eines entsprechenden Projektmerkmals an. Da Projektmerkmale sich nicht aufeinander beziehen können, lässt sich die Abhängigkeit zwischen *Ansprechpartner vor Ort* und *Ansprechpartner führt Abnahmen durch* nicht darstellen.

Handelt es sich bei dem Projekt um ein Auftragnehmerprojekt, so soll dieses durch die in Tabelle 7.5 aufgelisteten Eigenschaften charakterisiert sein.

Eigenschaft	Typ	Mögliche Werte
Art des Vertrags	genau einer	Dienstvertrag/Werkvertrag
Implementierungssprache	genau eine	C# / Java / C++
Entwicklungsstandort	mind. einer	München / Frankfurt (kein C++) / Stuttgart (kein Java)
24 Monate Wartungsvertrag	optional	Ja / Nein

Tabelle 7.5: Fiktive Projekteigenschaften bei Auftragnehmerprojekt für IT-Dienstleister

Im Szenario für Auftragnehmerprojekte nehmen wir an, dass das Unternehmen drei Entwicklungsstandorte hat, von denen an zwei Standorten nicht die Fachkräfte zur Durchführung von Projekten in allen drei der angebotenen Implementierungstechnologien zur Verfügung stehen (die Eigenschaft Implementierungssprache würde sich im Vorgehensmodell beispielsweise durch speziell abgestimmte Programmierrichtlinien und Vorgaben für die zum Einsatz kommende Werkzeuglandschaft auswirken): Wenn die Implementierungstechnologie C++ sein soll, kommt Frankfurt als Entwicklungsstandort nicht in Frage. Ebenso soll der Standort Stuttgart nicht wählbar sein, wenn die Implementierungssprache Java ist.

Sowohl die Eigenschaft *Implementierungssprache* als auch die Eigenschaft *Entwicklungsstandort* wären natürliche Kandidaten für die Abbildung durch Projektmerkmale im Tailoring des V-Modell XT. Um allerdings die Ausschlussbeziehung zwischen beiden Merkmalen darzustellen, bliebe nur die Einführung eines kombinierten Projektmerkmals mit den Merkmalswerten *C# in München*, *Java in München*, *C++ in München*, *C# in Frankfurt* usw. Während diese Abbildung zwar machbar ist, wäre die Anforderung, dass mindestens einer oder mehrere Standorte an dem Projekt beteiligt sein können, immer noch nicht abgebildet.

Wie die Anforderungen des fiktiven Beispiels mit dem feature-basierten Ansatz zum Tailoring dargestellt werden können, ist in Abbildung 7.17 dargestellt.

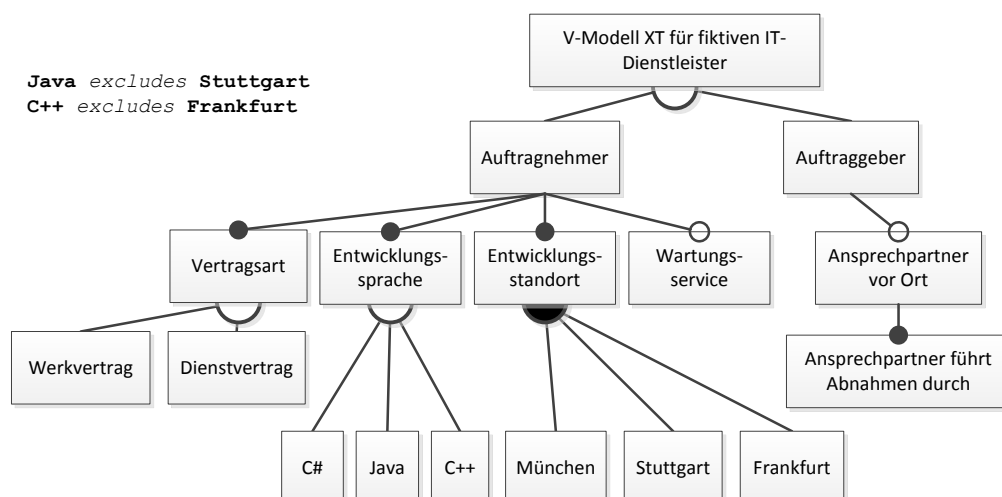


Abbildung 7.17: Fiktives Beispiel umgesetzt mit feature-basiertem Tailoring

Auch wenn das fiktive Szenario unvollständig ist, und die Anforderungen an das Tailoring in einer realen Situation sicher umfangreicher ausfallen würden, zeigen auch die wenigen Anforderungen deutlich, wo das klassische Tailoring des V-Modell XT an seine Grenzen stößt und wie das feature-basierte Tailoring dazu beitragen kann, das Tailoring eines Vorgehensmodells so zu spezifizieren, dass die Anforderungen korrekt umgesetzt werden.

7.3.5 Diskussion

In der hier beschriebenen Studie wurde der in der vorliegenden Arbeit entwickelte Ansatz zum feature-basierten Tailoring auf das V-Modell XT angewendet. Hierzu wurde das Metamodell des V-Modell XT entsprechend angepasst und

7.4 Experimente zur Evaluierung

ein Werkzeug entwickelt, welches Modellinhalte vom alten auf das neue Metamodelle migriert. Bevor das feature-basierte Tailoring im V-Modell XT vollständig nutzbar ist, müsste ein Werkzeug entwickelt werden, welches das feature-basierte Tailoring für den Vorgehensmodell-Anwender ermöglicht. Da der V-Modell XT Projektassistent für seine korrekte Funktion die Metamodellestrukturen des klassischen V-Modell XT Tailoring voraussetzt, lässt er sich nicht ohne weiteres auf die neuen Strukturen anpassen. Wie so ein Werkzeug aussehen könnte und wie es technisch zu realisieren wäre zeigt der *Process Suite Generator* zum SE Book (siehe Abschnitt 7.2.4).

Ein Ziel der Studie war, die Variabilitätspunkte im V-Modell XT so zu wählen bzw. anzupassen, dass die Mächtigkeit des feature-basierten Tailorings im V-Modell XT auch genutzt werden kann und ein feingranulares Tailoring – beispielsweise auf Basis einzelner Produkte – ermöglicht wird. Die Überlegungen hierzu haben ergeben, dass Vorgehensbausteine bereits der geeignete Variabilitätspunkt sind, um ein feingranulares Tailoring im V-Modell XT zu realisieren und in der Praxis des V-Modell XT bereits in dieser Art verwendet werden. Da die Variabilität des Modells beim Tailoring durch die Verwendung eines Konfigurationsmodells gegenüber der klassischen festen Struktur mit Projekttypen, Projekttypvarianten und Projektmerkmalen erhöht wird, kann allerdings vermutet werden, dass sich die Anzahl „kleiner“ Vorgehensbausteine mit einem oder wenigen Produkten deutlich vergrößern wird. Im oben beschriebenen fiktiven Beispiel etwa würden möglicherweise Vorgehensbausteine für die jeweilige Implementierungssprache erstellt, welche nur die Programmierrichtlinien zur jeweiligen Sprache und Vorgaben zur Werkzeuginfrastruktur beinhalten.

7.4 Experimente zur Evaluierung

In diesem Abschnitt beschreiben wir zwei Experimente, welche Eigenschaften des feature-basierten Tailoring durch Gegenüberstellung mit alternativen Tailoring-Ansätzen überprüfen und die Tailoring-Ansätze gegeneinander positionieren. Im Verlauf der vorliegenden Arbeit haben wir ein besonderes Augenmerk auf folgende Eigenschaften gelegt:

- Für den Prozessingenieur: Flexibilität und Ausdrucksmächtigkeit bei der Projektcharakterisierung mit Hilfe eines Konfigurationsmodells.
- Für den Anwender eines Vorgehensmodells:
 - Einfachheit und Transparenz bei der Durchführung des Tailorings.
 - Vergleichbarkeit der entstehenden Vorgehensmodell-Ausprägungen.
 - Nachvollziehbarkeit des Tailorings.

Die Experimente beschränken sich auf die Sicht der Vorgehensmodell-Anwender. Die Flexibilität und Ausdrucksmächtigkeit des feature-basierten Tailoring aus Sicht des Prozessingenieurs sind in der Fallstudie zum SE Book (in Abschnitt 7.2), in der Fallstudie zum V-Modell XT durch den Vergleich mit dem klassischen Tailoring dieses Vorgehensmodells (siehe Abschnitt 7.3) und in Kapitel 6 deutlich geworden.

7.4.1 Umfeld

Gegenstände beider Experimente sind das *Eclipse Process Framework (EPF)* [Ecl11a] mit dem dazugehörigen Werkzeug *EPF Composer*, das *V-Modell XT 1.4* [Wei12]

mit dem Werkzeug *V-Modell XT Projektassistent* und als Vertreter des feature-basierten Tailorings das *SE Book* mit dem begleitenden Werkzeug *Process Suite Generator*².

Beide Experimente erfolgen im akademischen Umfeld in einem kontrollierten Rahmen. Im ersten Experiment führen 10 Studenten im Rahmen einer universitären Vorlesung eine projektspezifische Anpassung basierend auf einem fiktiven Projektszenario durch. Von den Mitgliedern dieser Gruppe kann angenommen werden, dass sie mit allen drei Tailoring-Ansätzen gleichermaßen (nicht) vertraut sind. Im zweiten Experiment führen 6 Teilnehmer aus dem akademischen Umfeld eine projektspezifische Anpassung basierend auf einem komplexeren, realen Projektszenario durch. Die Mitglieder dieser Gruppe sind Experten für dem jeweils verwendeten Tailoring-Ansatz.

Um die beiden Experimente in den folgenden Ausführungen zu unterscheiden, verwenden wir für das erste Experiment mit Studenten die Bezeichnung **Experiment mit Studenten** und für das zweite Experiment mit Experten die Bezeichnung **Experiment mit Experten**.

Im Experiment mit Studenten verwenden wir als Fallbeispiel zum Tailoring ein einfaches, fiktives Projektszenario. Im Experiment mit Experten verwenden wir einen Projektsteckbrief eines realen Projekts. Dieser stammt aus der Datensammlung der im Rahmen von [MFWL⁺12] durchgeführten Untersuchungen.

7.4.2 Ziele und Forschungsfragen

Das Ziel der hier beschriebenen Experimente ist die Überprüfung der Eigenschaften des feature-basierten Tailorings aus Sicht eines Vorgehensmodell-Anwenders, d.h. einer das Tailoring durchführenden Person. Eine Gegenüberstellung der Tailoring-Ansätze aus Sicht des Prozessingenieurs erfolgt im Rahmen dieses Experiments nicht. Die Möglichkeiten zur Beschreibung von Tailoring-Profilen mit Hilfe des feature-basierten Tailorings wurden aber in der Studie zum *SE Book* anhand der Inhalte zur Demonstration (siehe Abschnitt 7.2) und in der Studie zum *V-Modell XT* anhand des fiktiven Beispiels (siehe Abschnitt 7.3) verglichen.

Mit den Experimenten sollen folgende Forschungsfragen beantwortet werden:

Forschungsfrage 1: Wie wird die Durchführung des Tailorings empfunden?

1. Ist das Tailoring einfach durchzuführen?
2. Sind die beim Tailoring zu treffenden Entscheidungen klar?
3. Sind die Auswirkungen der Entscheidungen auf das Vorgehensmodell transparent?
4. Ist das Tailoring auch für einen nicht mit der inneren Struktur des Vorgehensmodells vertrauten Anwender durchzuführen?

Forschungsfrage 2: Wie gut lassen sich die in Form eines Projektsteckbriefs beschriebenen Projektcharakteristika umsetzen/abbilden und wie gut „passt“ das Tailoring-Ergebnis zu der in Form des Projektsteckbriefs beschriebenen Projektcharakterisierung?

Forschungsfrage 3: Ist das Tailoring nachvollziehbar bzw. reproduzierbar, d.h. erlaubt das Tailoring-Ergebnis Rückschluss auf die beim Tailoring getroffenen Entscheidungen (auf die „Rationale“ des Tailorings)?

² auf dem Stand, den das Modell und das Werkzeug zum Zeitpunkt der Beendigung der Studie in Abschnitt 7.2 hatten.

7.4 Experimente zur Evaluierung

Die Gegenüberstellung der Antworten auf diese Fragen zu jedem der drei Untersuchungsgegenstände ergibt einen Eindruck der Stärken und Schwächen der drei Ansätze.

Die Experimente liefern Hinweise auf die im Laufe dieser Dissertation postulierten positiven Eigenschaften des feature-basierten Tailorings. Sie stellen damit eine Grundlage dar für weitere Untersuchungen, bei welchen die Fähigkeiten und Grenzen des feature-basierten Tailorings ausgelotet werden (siehe auch Abschnitt 8.3). Das Vorgehen der hier beschriebenen Experimente bietet sich als Vorlage an für weiterführende und umfassendere Untersuchungen.

7.4.3 Studienaufbau

Im Aufbau ähneln sich beide Experimente. Der Übersichtlichkeit halber beschreiben wir den Aufbau der Experimente im Folgenden dennoch unabhängig voneinander.

Experiment mit Studenten

Den teilnehmenden Studenten wird ein Tailoring-Szenario vorgegeben, welches sie in jedem der drei Werkzeuge (soweit möglich) auf das Vorgehensmodell abbilden sollten. Das fiktive Szenario ist durch folgende Projekteigenschaften charakterisiert:

- Anwendungstyp: Webanwendung
- Projektumfang: klein (ca. 8 PM)
- Implementierungssprache: Java
- Verwendung von Kaufkomponenten: Ja, Datenbank
- Vorgehensweise: Timeboxing

Die Projekteigenschaften sind bewusst so vorgegeben, dass sie in keinem der drei betrachteten Vorgehensmodelle und dem dazugehörigen Werkzeug eins zu eins abzubilden sind. Dadurch soll vermieden werden, dass die Wahl des Szenarios Einfluss nimmt auf die Meinung der Studenten zu den einzelnen Werkzeugen.

Anschließend füllen die Studenten einen Fragebogen zu den drei Werkzeugen aus. Die Fragen im Fragebogen sind:

- Ich fand mich in dem Werkzeug schnell zurecht.
- Ich wusste jederzeit, wie ich die Projekteigenschaften im Tailoring umsetzen kann.
- Die Menge an Entscheidungen ist angemessen.
- Die Konsequenzen meiner Entscheidungen waren mir immer klar.
- Ich wurde bei der Dokumentation meiner Entscheidungen gut unterstützt.
- Das Tailoring ist auch für nicht mit dem Modell vertraute Mitarbeiter einfach.

Alle Fragen sind auf einer Likert-Skala (mit dem Wertebereich: Stimmt gar nicht, Stimmt wenig, Stimmt teils-teils, Stimmt ziemlich, Stimmt völlig) zu beantworten.

Zudem werden die Studenten befragt, inwieweit die vorgegebenen Projekteigenschaften mit jedem der drei Vorgehensmodelle bzw. Tailoring-Werkzeuge umsetzbar sind.

Experiment mit Experten

Die zentrale Idee des Experiments ist, die betrachteten Tailoring-Verfahren für ein und das selbe (anonymisierte) Projektszenario durchzuführen und dabei die Stärken und Schwächen des jeweiligen Ansatzes herauszuarbeiten.

Das Vorgehen des Experiments ist in Abbildung 7.18 (für einen der drei Tailoring-Ansätze) dargestellt.

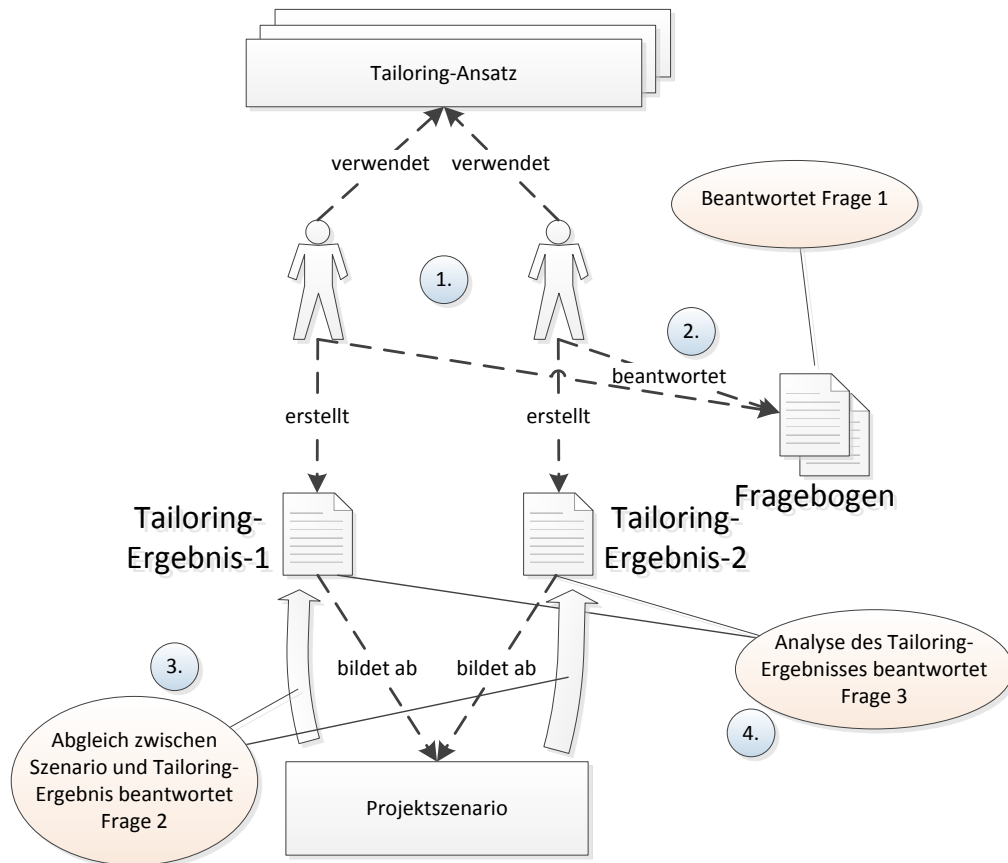


Abbildung 7.18: Experiment zur Evaluierung des feature-basierten Tailorings

Das Experiment wird mit 6 Experten durchgeführt. Jeweils zwei Experten verwenden unabhängig voneinander einen Tailoring-Ansatz. Konkret verwenden zwei Experten zur Erstellung eines projektspezifisch angepassten Vorgehensmodells das *Eclipse Process Framework*, zwei verwenden das *V-Modell XT* und zwei verwenden das *SE Book* (als Vertreter des feature-basierten Tailorings). Im folgenden skizzieren wir, wie durch das Vorgehen in Abbildung 7.18 die Forschungsfragen beantwortet werden.

Im ersten Schritt (vergleiche Abbildung 7.18) führen die Experten eine projektspezifische Anpassung gemäß des anonymisierten Projektsteckbriefs durch. Sie verwenden dabei den ihnen zugedachten Tailoring-Ansatz.

Forschungsfrage 1: Alle Experten beantworten nach der Erstellung der projektspezifischen Ausprägung des jeweiligen Vorgehensmodells einen Fragebogen (Schritt 2 in Abbildung 7.18). Mit dem Fragebogen wird insbesondere die Forschungsfrage 1 (siehe Abschnitt 7.4.2) adressiert.

Forschungsfrage 2: Zur Beantwortung der Frage 2 gibt jeder Experte vor Durchführung des Tailorings seine Einschätzung ab, wie viele Charakteristika des in

7.4 Experimente zur Evaluierung

Form des Projektsteckbriefs beschriebenen Projekts für das Tailoring des Vorgehensmodells relevant sind. Nach der Durchführung des Tailorings reflektiert der Experte, wie viele der Eigenschaften er mit Hilfe des Tailoring-Ansatzes hat abbilden können (siehe Schritt 3 in Abbildung 7.18).

Forschungsfrage 3: Zur Beantwortung dieser Frage wird das Tailoring-Ergebnis auf die Existenz von Artefakten hin überprüft, welche das Tailoring und die dabei getroffenen Entscheidungen dokumentieren (in Abbildung 7.18 entspricht dies Schritt 4).

Der vollständige Fragebogen befindet sich in Anhang B.

Wahl des Projektszenarios: Als Basis für das Tailoring mit allen drei Ansätzen dient ein Projektszenario, welches in Form eines Projektsteckbriefs an die Experten verteilt wird. Um durch die Wahl des Projektszenarios nicht einen der drei Tailoring-Ansätze zu bevorzugen, wurde folgendes Vorgehen gewählt:

- Der Projektsteckbrief wird unverändert aus der Datenbasis zu [MFWL⁺12] übernommen. Lediglich sensible Informationen, wie Details zum Kunden, werden unkenntlich gemacht.
- Der Projektsteckbrief wird von einer mit den Eigenschaften der einzelnen Tailoring-Ansätze nicht vertrauten Person aus den in [MFWL⁺12] gesammelten Projektsteckbriefen (zufällig) gewählt.

Auswahl der Experten: Die Experten zur Durchführung des Experiments kommen aus dem akademischen Umfeld. Jeder der Experten ist mit dem entsprechenden Tailoring-Ansatz (und dem dabei zum Einsatz kommenden Werkzeug) vertraut. Dadurch stellen wir sicher, dass die Ergebnisse nicht durch Unkenntnis des Tailoring-Ansatzes und die damit verbundene Einarbeitungshürde beeinflusst werden.

7.4.4 Ergebnisse und Diskussion

Im folgenden Abschnitt fassen wir die Ergebnisse beider Experimente zusammen. Der Übersichtlichkeit halber beschreiben wir die Ergebnisse beider Experimente getrennt voneinander.

Experiment mit Studenten

Das Laborexperiment zum Vergleich der Tailoring-Werkzeuge Process Suite Generator (mit feature-basiertem Tailoring)³, EPF Composer und V-Modell XT Projektassistent wurde mit 10 Studenten durchgeführt. Aufgrund der geringen Teilnehmerzahl konnten die Ergebnisse nicht statistisch ausgewertet werden. Da die Studenten vor Durchführung des Experiments mit allen drei Tailoring-Werkzeugen gleichermaßen (nicht) vertraut waren, geben die Ergebnisse des Laborexperiments durchaus Hinweise auf die Verständlichkeit und Bedienbarkeit der verschiedenen Tailoring-Ansätze. Drei zentrale Ergebnisse des Fragebogens zum Laborexperiment sind in Abbildung 7.19 dargestellt.

Der EPF Composer wurde von den Teilnehmern des Laborexperiments als das Werkzeug mit der größten Einstiegshürde wahrgenommen. Der Process Suite Generator und der V-Modell XT Projektassistent wurden beide als Werkzeuge wahrgenommen, in denen man sich gut zurechtfindet. Auch die Einschätzung der Teilnehmer zum Umgang mit dem Werkzeug, wenn man mit diesem

³ auf dem Stand, den er nach der Fallstudie aus Abschnitt 7.2 hatte.

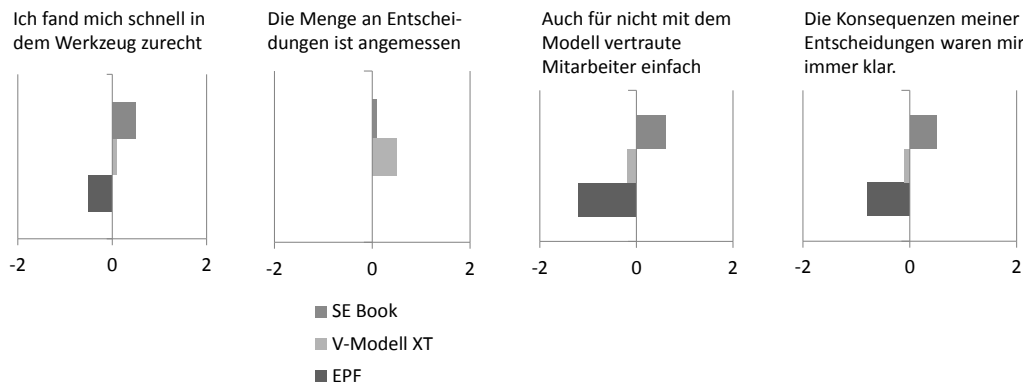


Abbildung 7.19: Ergebnisse des Laborexperiments zum Tailoring im Process Suite Generator, dem EPF Composer und dem V-Modell XT Projektassistent

nicht vertraut ist, zeigt dieselbe Tendenz. Der leicht negative Ausschlag bei der Einschätzung des V-Modell XT Projektassistenten lässt sich möglicherweise damit erklären, dass nicht ohne weiteres ersichtlich ist, welche Auswirkungen die Wahl der verschiedenen Projekttypen, Projekttypvarianten und Projektmerkmale auf die projektspezifische Ausprägung des Vorgehensmodells haben. Die Meinung der Teilnehmer zur Aussage „Die Konsequenzen meiner Entscheidungen waren mir immer klar“ scheinen diesen Verdacht zu bestätigen.

Bei der Frage nach der Anzahl der beim Tailoring zu treffenden Entscheidungen schnitt das V-Modell XT mit seinen höchstens $2 + n^4$ Entscheidungen in der Gunst der Teilnehmer am besten ab. Die Anzahl von Entscheidungen beim Tailoring des SE Book hängt nur von der Größe des Konfigurationsmodells ab und ist damit prinzipiell beliebig. Da das Vorgehensmodell im EPF Composer aus einer Reihe von Methodenbausteinen erst zusammengesetzt werden muss, ist die Anzahl von Entscheidungen im EPF Composer nicht eindeutig zu bestimmen.

Experiment mit Experten

In der zweiten Spalte der Tabelle 7.6 ist die Anzahl der Projektcharakteristika dargestellt, die *vor* Durchführung des Tailorings von den Teilnehmern identifiziert werden konnten. Die dritte Spalte zeigt die Zahl der Charakteristika, welche *nach* dem Tailoring als relevant für die Anpassung des Vorgehensmodells betrachtet wurden. Die letzte Spalte stellt die Anzahl der Charakteristika dar, die durch Verwendung des jeweiligen Ansatzes im Tailoring umgesetzt werden konnten. Für jeden der Ansätze ist jeweils der Mittelwert der beiden, unabhängig voneinander durchgeführten, projektspezifischen Anpassungen dargestellt. Die starke Streuung der Zahlen in ihrer absoluten Höhe ist mit dem Einfluss der Experten (bedingt durch jeweilige Vorkenntnisse und persönliche Herangehensweise) zu erklären. Mit dem anzupassenden Vorgehensmodell im Hinterkopf nannten die Teilnehmer, welche das Experiment für EPF durchführten, allgemeine *Themenbereiche*, wie etwa Rollen, Guidance, Artefakte und Prozessmodell. Mit dem SE Book bereits im Hinterkopf antworteten die entsprechenden Experten wesentlich differenzierter. Die Antwort eines der Teilnehmer für das SE Book auf die Frage der für das Tailoring relevanten Passagen im Projektsteckbrief war: „Prinzipiell ist alles relevant“. Auch unterschieden sich die

⁴ Zwei für Projekttyp und Projekttypvariante plus n für die Anzahl der Projektmerkmale

7.4 Experimente zur Evaluierung

Ergebnisse abhängig davon, wie „buchstabengetreu“ ein Teilnehmer das Tailoring interpretierte. Während ein Teilnehmer von den identifizierten Charakteristika nur etwa die Hälfte *exakt* auf entsprechende Kriterien im Tailoring abbilden konnte, kommentierte der andere Teilnehmer: „Indirekt lassen sich alle identifizierten Eigenschaften des Projekts abbilden“. Auch die Experten zum V-Modell XT ließen sich durch die Kenntnis des Tailorings leiten und identifizierten überwiegend Charakteristika im Projektsteckbrief, die ihnen aus dem Tailoring des V-Modell XT bereits bekannt waren.

Ansatz	Vorher	Nachher	Umgesetzt	% Umgesetzt
V-Modell XT	6	9,5	7,5	ca. 80 %
SE Book	18	40	40	indirekt 100 % (siehe Text)
EPF	3,5	4	3,5	ca. 88 %

Tabelle 7.6: Überblick über Projektcharakteristika

Tabelle 7.7 fasst die Einschätzung der Teilnehmer des Experiments zum jeweils verwendeten Tailoring-Ansatz zusammen. Dabei wurde zu den folgenden Fragen die Zustimmung abgefragt⁵:

Frage 4.a. Mir war jederzeit klar, *wie ich beim Tailoring vorzugehen* hatte.

Frage 4.b. Ich wusste, *welche Entscheidungen* ich zu treffen hatte.

Frage 4.c. Mir war jederzeit ersichtlich, *wie sich meine Entscheidungen* auf die Inhalte des Vorgehensmodells *auswirken*.

Die Zustimmung wurde im Fragebogen auf einer Likert-Skala mit folgenden Werten beantwortet:

- Stimme sehr zu (+2)
- Stimme zu (+1)
- Weder/Noch (0)
- Stimme nicht zu (-1)
- Stimme gar nicht zu (-2)

In Tabelle 7.7 ist zu jeder Frage und zu jedem Ansatz jeweils der Mittelwert der numerischen Werte aus den beiden unabhängig voneinander durchgeführten Anpassungen dargestellt.

Frage	V-Modell XT	SE Book	EPF
4.a.	2 (stimme sehr zu)	1 (stimme zu)	1,5 (stimme sehr zu)
4.b.	2 (stimme sehr zu)	0,5 (Stimme zu)	0,5 (Stimme zu)
4.c.	1 (stimme zu)	1,5 (stimme sehr zu)	0 (weder/noch)

Tabelle 7.7: Einschätzung zur Durchführung des Tailorings

Die Ergebnisse in Tabelle 7.7 spiegeln die Erwartungen an das Experiment wieder: das Tailoring des V-Modell XT erlaubt nur vergleichsweise wenig Möglichkeiten zur Konfiguration der projektspezifischen Ausprägung des Vorgehensmodells. Dementsprechend klar sind die zu treffenden Entscheidungen und die Vorgehensweise.

⁵ Die Nummerierung der Fragen entspricht der Nummerierung im Fragebogen in Anhang B.

Überraschend ist die Einschätzung der Studienteilnehmer, dass die Auswirkung des Tailorings auf das Vorgehensmodell bei EPF nicht offensichtlich sei. Da das Tailoring bei EPF im Wesentlichen in einer Neuerstellung bzw. Zusammenstellung der Inhalte des Vorgehensmodells besteht (vergleiche Abschnitt 3.2.2), wäre hier ein besseres Abschneiden zu erwarten gewesen. Begründet wurde die Einschätzung von den Teilnehmern mit der Komplexität des Werkzeugs zum Tailoring (dem EPF Composer) und der Komplexität des Eclipse Process Framework selbst. Aufgrund der Komplexität sei nicht immer klar, wo und wie sich Anforderungen an das Vorgehensmodell umsetzen lassen und welche Konsequenzen die jeweiligen Realisierungsalternativen haben.

Da im Tailoring des SE Book – ähnlich wie beim V-Modell XT – die Entscheidungen zum Tailoring durch das Konfigurationsmodell des feature-basierten Tailoring vorgegeben sind, wäre auf die Frage nach der Klarheit der Entscheidungen ein positiveres Ergebnis zu erwarten gewesen. Die Ursache für das Ergebnis ist in dem beim Experiment zum Einsatz gekommenen Konfigurationsmodell zu suchen: hierbei handelte es sich um das in Zusammenarbeit mit der T-Systems International GmbH für den produktiven Einsatz in diesem Unternehmen entwickelte Konfigurationsmodell zum SE Book (vergleiche Abschnitt 7.2). Es enthält zahlreiche Projektcharakteristika, welche speziell mit den Besonderheiten des Unternehmens zu tun haben. Diese Kriterien waren für die Teilnehmer des Experiments entweder nicht zu beantworten oder sorgten zumindest für Verwirrung.

In Tabelle 7.8 befinden sich die Antworten der Teilnehmer auf die Frage, ob ihnen das Tailoring leicht gefallen sei. Folgende Antworten waren dabei möglich:

- sehr leicht gefallen (+2)
- leicht gefallen (+1)
- Weder/Noch (0)
- schwer gefallen (-1)
- sehr schwer gefallen (-2)

Ansatz	Ergebnis
V-Modell XT	1 (leicht gefallen)
SE Book	1 (leicht gefallen)
EPF	-0,5 (etwas schwer gefallen)

Tabelle 7.8: Ergebnisse zu Frage 7.a.

Die Ergebnisse in Tabelle 7.8 liegen voll und ganz im Rahmen der Erwartungen und lassen sich mit dem Aufwand begründen, der mit dem Tailoring im jeweiligen Ansatz verbunden ist.

Ob der Tailoring-Ansatz und das dabei zu verwendende Werkzeug nach Einschätzung der Teilnehmer Vorkenntnisse und Einarbeitung benötigt, war Gegenstand der letzten Frage im Fragebogen. Mögliche Antworten hierbei waren:

- Kaum Einarbeitung (+4)
- Etwas Einarbeitung (+3)
- Viel Einarbeitung (+2)
- Sehr viel Einarbeitung (+1)

Die Ergebnisse zu dieser Frage befinden sich in Tabelle 7.9.

7.4 Experimente zur Evaluierung

Ansatz	Ergebnis
V-Modell XT	3,5 (kaum bis etwas Einarbeitung)
SE Book	4 (kaum Einarbeitung)
EPF	1,5 (viel bis sehr viel Einarbeitung)

Tabelle 7.9: Ergebnisse zu Frage 7.b.

Wie die Ergebnisse in Tabelle 7.8 sind auch die Ergebnisse auf die Frage nach dem Einarbeitungsaufwand in Tabelle 7.9 kaum überraschend. Der EPF Composer ist ohne tiefere Kenntnisse kaum zu bedienen, während beim V-Modell XT Projektassistent und beim Process Suite Generator (für das SE Book) in der Regel der Hinweis auf die entsprechende Eingabemaske zum Tailoring genügt.

Im Hinblick auf die Forschungsfragen zeichnet das Experiment folgendes Bild:

Forschungsfrage 1: Wie wird die Durchführung des Tailoring empfunden? Einfachheit, Einarbeitungsaufwand und Nachvollziehbarkeit werden beim V-Modell XT und beim SE Book etwa gleich eingeschätzt. Das Tailoring in EPF wird als komplexer empfunden, was aber angesichts des bei EPF verfolgten Ansatzes zur Anpassung des Vorgehensmodells kaum überraschen dürfte. Die anfängliche Lernkurve bei EPF und den begleitenden Werkzeugen ist so hoch, dass die Durchführung einer projektspezifischen Anpassung nur von einem EPF-Experten erwartet werden kann.

Forschungsfrage 2: Wie gut lassen sich die in Form eines Projektsteckbriefs beschriebenen Projektcharakteristika umsetzen/abbilden? Das Experiment ergibt bei allen drei Ansätzen eine (ähnlich) hohe Überdeckung zwischen im Projektsteckbrief identifizierten und im Tailoring umgesetzten Charakteristika. Beim Entwurf des Experiments wurde vom Autor die „Färbung“ der Ergebnisse durch die Vorkenntnisse der Studienteilnehmer in Bezug auf den jeweils verwendeten Ansatz unterschätzt. Die Frage nach den für das Tailoring relevanten Projektcharakteristika wurde von allen Teilnehmern „durch die Brille“ des jeweiligen Ansatzes gesehen und interpretiert. Die Teilnehmer für EPF fassten die für das Tailoring relevanten Projekteigenschaften als *Themengebiete* für die Umsetzung in EPF auf, während die Teilnehmer für das SE Book sehr differenziert Charakteristika identifizierten und laut Aussage eines der Teilnehmer potenziell „alles“ im Steckbrief als relevant für das Tailoring betrachteten. Dementsprechend signifikant unterscheiden sich die absoluten Zahlen der identifizierten (und umgesetzten) Charakteristika in den drei Ansätzen. Eine indirekte Erkenntnis des Experiments ist daher offenbar, dass der Tailoring-Ansatz selbst beim Anwender den Erkenntnisgewinn über die projektspezifische Anpassung eines Vorgehensmodells fördern kann (vergleiche hierzu die Ergebnisse in Tabelle 7.6).

Forschungsfrage 3: Ist das Tailoring nachvollziehbar bzw. reproduzierbar? Diese Frage wurde nicht von den Teilnehmern des Experiments beantwortet, sondern durch Analyse der drei untersuchten Tailoring-Ansätze. Im V-Modell XT Projektassistent lässt sich zu jedem Projektmerkmal eine Begründung angeben. Wenn die projektspezifische V-Modell XT Projektdatei mit den Unterlagen des Projekts abgelegt wird, ist damit das Tailoring und die dabei getroffenen Entscheidungen nachvollziehbar. Auch im SE Book können Begründungen zu den beim Tailoring getroffenen Entscheidungen angegeben werden (je nach Entwurf des Konfigurationsmodells können manche Begründungen auch verpflichtend

sein). Diese werden bei Generierung des projektspezifisch angepassten SE Book in einen *Tailoring Report* aufgenommen, welcher alle Kriterien mit ihrer Belegung und der dazugehörigen Begründung auflistet. Bei einem projektspezifischen Vorgehensmodell basierend auf EPF ist die Entstehungsgeschichte nicht ohne weiteres zurückzuverfolgen. Für die Nachvollziehbarkeit und Reproduzierbarkeit ist man bei EPF darauf angewiesen, dass der Prozessingenieur seine Entscheidungen und die dazu führenden Überlegungen selbstständig dokumentiert.

Die Ergebnisse des Experiments spiegeln im Wesentlichen wieder, was durch Analyse und Einschätzung der drei betrachteten Tailoring-Ansätze zu erwarten war (vergleiche hierzu insbesondere Kapitel 3 und 4). Insofern kann vermutet werden, dass die Ergebnisse auch bei Wiederholung mit verschiedenen Projektsteckbriefen und mit einer größeren Anzahl von Teilnehmern in der Tendenz unverändert blieben. Interessant wäre eine Wiederholung des Experiments mit einer unternehmensunabhängigen Implementierung des feature-basierten Tailorings. Es kann spekuliert werden, dass in diesem Fall die Ergebnisse deutlicher zugunsten des feature-basierten Tailorings ausfallen würden.

7.4.5 Beeinträchtigung der Validität

Im folgenden Abschnitt fassen wir zusammen, welche Faktoren die Validität der Ergebnisse beeinträchtigen können.

Beide hier beschriebenen Experimente wurde mit einer relativ kleinen Teilnehmerzahl im rein akademischen Umfeld durchgeführt. Dementsprechend sind bei der Betrachtung und Wertung der Ergebnisse folgende Aspekte zu berücksichtigen:

Projektszenario: Die projektspezifische Anpassung mit Hilfe der drei Tailoring-Ansätze erfolgt in beiden Experimenten *für nur ein Projektszenario*. Durch die in den Projektszenarien beschriebenen Projektcharakteristika könnte einer der drei Tailoring-Ansätze bevorzugt sein. Diesem Risiko wurde begegnet, indem der Projektsteckbrief beim Experiment mit Experten von einer unbeteiligten Person aus einer Menge von drei Steckbriefen gewählt wurde. Im Experiment mit Studenten wurde ein einfaches, fiktives Szenario vorgegeben, welches in keinem der drei Ansätze eins zu eins umsetzbar war. Um das Risiko einer ungewollten Bevorzugung eines der drei Ansätze gänzlich auszuschließen, müsste das Experiment in einer anknüpfenden Studie für eine Reihe unterschiedlicher Projektsteckbriefe wiederholt werden.

Teilnehmer: Die Experten stammen aus dem akademischen Umfeld und sind alle persönlich mit dem Autor dieser Dissertation bekannt. Es kann nicht gänzlich ausgeschlossen werden, dass die Antworten im Fragebogen durch ein „Wohlwollen“ der Experten gegenüber dem Autor beeinflusst sind. Um dieses Risiko zu minimieren, wurden die Experten – soweit möglich – über die Hintergründe und Absichten des Experiments im Dunkeln gelassen. Auch den Teilnehmern beim Experiment mit Studenten wurden die Hintergründe und Absichten nicht kommuniziert.

Ein weiteres Risiko für die Validität der Ergebnisse erwächst aus der geringen Anzahl der Teilnehmer. In einer anknüpfenden Studie müsste das Experiment mit einer größeren Zahl von Teilnehmern durchgeführt werden, bei denen zudem das Risiko einer Verfälschung der Ergebnisse durch persönliche Verbundenheit ausgeschlossen werden kann.

7.5 Zusammenfassung

Vergleichbarkeit der Ansätze: Als Ausgangsbasis für die Anpassung eines auf EPF basierenden Vorgehensmodells wählten beide betroffenen Experten den OpenUP [Ecl11b]. Bei diesem handelt es sich – genau wie beim V-Modell XT – um einen unternehmensneutralen, *generischen* Vorgehensstandard. Demgegenüber stand das SE Book der T-Systems International GmbH als Vertreter des feature-basierten Tailorings. Das Tailoring dieses Vorgehensmodells enthält einige Besonderheiten, welche der Abstimmung des Vorgehensmodells auf das Unternehmen und sein Projektportfolio geschuldet sind. Die Ergebnisse des Experiments sind daher möglicherweise beeinflusst von dem Umstand, dass einige Tailoring-Entscheidungen im SE Book mit den im Projektsteckbrief zur Verfügung stehenden Informationen nicht entscheidbar waren. Um dieses Risiko zu minimieren, müsste das Experiment erneut durchgeführt werden mit einer unternehmensunabhängigen Umsetzung des feature-basierten Tailorings.

7.5 Zusammenfassung

In diesem Kapitel haben wir zwei Fallstudien beschrieben, mit denen wir die Tragfähigkeit des feature-basierten Tailorings an realen Vorgehensmodellen überprüft und nachgewiesen haben.

Die Fallstudie zum SE Book wurde durchgeführt im Rahmen einer Forschungs-kooperation zwischen den Telekom Laboratories, der T-Systems International GmbH und der Technischen Universität München⁶. Neben den technischen Voraussetzungen auf Ebene des Vorgehensmodells (Definition eines Metamodells für das Konfigurationsmodell, Vereinbarung der Operationen und der variablen Modellinhalte) ist im Rahmen der Forschungskoope-ration ein vollwertiges Software-Werkzeug zur Durchführung des feature-basierten Tailorings entstanden, welches bei der T-Systems International GmbH für alle Software- und Systementwicklungsprojekte eingesetzt wird.

Mit der Fallstudie zum V-Modell XT haben wir nachgewiesen, dass das feature-basierte Tailoring sich auch in diesem, von der Architektur und dem inneren Aufbau stark vom SE Book abweichenden, Vorgehensmodell umsetzen lässt. Durch das Beispiel zur Evaluierung wurde zudem deutlich, welchen Mehrwert die Umsetzung des feature-basierten Tailorings im V-Modell XT bietet. Da bei dieser Fallstudie der Nachweis der Machbarkeit im Vordergrund stand, wurde kein Werkzeug zur Durchführung des feature-basierten Tailorings im V-Modell XT entwickelt. Ein Software-Werkzeug zu diesem Zweck könnte aber ähnlich wie der *Process Suite Generator* für das SE Book gestaltet sein.

Im dritten Teil dieses Kapitels haben wir zwei im akademischen Umfeld durchgeführte, kontrollierte Experimente zum Vergleich der Tailoring-Ansätze des Eclipse Process Framework, des V-Modell XT und des SE Book (als Vertreter des feature-basierten Tailorings) beschrieben. Der Fokus beider Experiment lag auf den Eigenschaften der Tailoring-Ansätze aus Sicht von Vorgehensmodell-Anwendern, also das Tailoring durchführenden Personen. Die Ergebnisse der Experimente bestätigen die Analyse der Anpassungsfähigkeit von Vorgehensmodellen in Kapitel 3, die theoretischen Überlegungen zu den Eigenschaften des feature-basierten Tailorings in Kapitel 5 und die Ergebnisse der beiden Fallstudien in den Abschnitten 7.2 und 7.3.

⁶ vertreten durch den Lehrstuhl Software & Systems Engineering an der Fakultät für Informatik.

Zusammenfassung und Ergebnisse

In diesem Kapitel blicken wir zurück auf die Ausgangslage, die Probleme bei aktuell existierenden Ansätzen zur projektspezifischen Anpassung von Vorgehensmodellen und den mit dieser Arbeit vorgeschlagenen Lösungsansatz. Wir rekapitulieren die zentralen Ergebnisse und legen dar, welche anknüpfenden Untersuchungen nötig und sinnvoll erscheinen.

Am Ende dieses Kapitels wird der Leser die zentralen Inhalte und Ergebnisse dieser Dissertation rekapituliert haben. Anknüpfend an den Ergebnissen dieser Arbeit wird er einen Eindruck offener Forschungsfragen und möglicher zukünftiger Arbeiten haben.

Übersicht

8.1	Ausgangslage	206
8.2	Lösung und Ergebnisse	207
8.3	Ausblick und offene Forschungsfragen	209

8.1 Ausgangslage

Vorgehensmodelle sind ein anerkanntes Mittel in der Software- und Systementwicklung, die Erfolgswahrscheinlichkeit eines Projekts hinsichtlich der Zeit-, Budget- und Qualitätsvorgaben zu steigern. Seit der Vorstellung eines einfachen Phasenmodells im Jahr 1956 und des Wasserfallmodells im Jahr 1970 wurden eine Vielzahl verschiedener Vorgehensweisen zur Strukturierung von Software- und Systementwicklungsprojekten vorgeschlagen. Gemein haben alle Vorgehensweisen, dass sie zur Anwendung an die spezifische Situation des Projekts angepasst werden müssen. Die Anpassung einer Vorgehensweise an die Situation eines Projekts wird als Tailoring bezeichnet.

Thema dieser Arbeit ist die Entwicklung von Mechanismen, die dem Ersteller eines Vorgehensmodells die Beschreibung der Kriterien zum Tailoring und deren Wirkung auf die Inhalte des Vorgehensmodells erlauben und welche dem Anwender eines Vorgehensmodells ermöglichen, das Tailoring transparent, nachvollziehbar und die Regeln des Vorgehensmodells respektierend durchzuführen.

8.1 Ausgangslage

Wie in Kapitel 1 ausgeführt, herrscht in Literatur, Forschung und Praxis Einigkeit über die Notwendigkeit der projektspezifischen Anpassung von Vorgehensmodellen. Viele Arbeiten befassen sich mit den Einflussfaktoren, die auf ein Projekt und damit auf sein Vorgehensmodell wirken. Eine eingehende Diskussion derartiger Arbeiten war Gegenstand des Kapitels 2. Situationsfaktoren, welche auf die Wahl eines Vorgehensmodells als solches und auf dessen Ausgestaltung Einfluss nehmen können, sind in der Regel informell beschrieben und es bleibt in der Regel unklar, wie erstens die relevanten Faktoren zu bestimmen sind und zweitens, wie genau diese auf das Vorgehensmodell wirken. „The main problems with contingency frameworks, such as Multiview, is that they lack a theoretical foundation for the environment factors and above all do not supply guidelines for the selection of method fragments on the basis of these factors“ [tV97].

In Kapitel 3 wurden die Möglichkeiten zur Anpassung bekannter Vorgehens-(Meta-)modelle beleuchtet. Hierbei lassen sich grob drei Kategorien unterscheiden:

- Für ein konkretes Projekt wird nach festgelegten Regeln eine neue, projektspezifische Vorgehensmodellausprägung *konstruiert*. Diese wird beispielsweise aus in einer Sammlung (*Method Base*) enthaltenen Bausteinen (*Method Fragments*) zusammengesetzt (vergleiche Abschnitte 2.5.1, 3.2.2 und 4.3).
- Das Vorgehensmodell wird projektspezifisch *konfiguriert*. Prominente Vertreter dieser Kategorie sind das V-Modell XT und in Ansätzen die schweizerische Projektführungsmethode HERMES (vergleiche Abschnitte 2.5.3, 3.2.1, 3.2.3 und 4.2).
- Die Vorgehensweise ist informell beschrieben. In diesem Fall sind auch die Vorgaben oder Empfehlungen zur Erstellung einer projektspezifischen Variante informell beschrieben. Agile Vorgehensweisen sind typische Vertreter dieser Kategorie (vergleiche Abschnitte 3.3 und 4.4).

Ansätze der ersten Kategorie umfassen die Konzepte zur Bildung eines projektspezifischen Vorgehensmodells aus Bausteinen und zur Definition der Bausteine selbst. Die Situationsbeschreibung eines Projekts ist in diesen Ansätzen nicht

modelliert. Sie bieten damit keine Hilfestellung, *nach welchen Kriterien und Regeln* Bausteine auszuwählen und zu kombinieren sind. Die Bildung eines Vorgehensmodells hängt wesentlich von der Expertise des Prozessingenieurs ab. Zudem ist ihre Entstehungsgeschichte – sofern nicht außerhalb der Vorgaben der Vorgehensweise dokumentiert – nicht transparent und nachvollziehbar und erschwert die Vergleichbarkeit der resultierenden projektspezifischen Vorgehensweisen – und damit der Projekte.

In Ansätzen der zweiten Kategorie sind die Regeln zur Bildung einer projektspezifischen Ausprägung des Vorgehensmodells direkt mit den konfigurierbaren Inhalten verbunden. *Das Modell selbst* beinhaltet die Informationen zur eigenen projektspezifischen Anpassung. Das Modell der Projektcharakterisierung in diesen Ansätzen ist aber so einfach gehalten, dass sich damit nur grobe Kriterien zur Beschreibung der Projektsituation darstellen lassen. Die Möglichkeiten und Grenzen des Tailoring im V-Modell XT haben wir im Abschnitt 3.2.1 und in der Fallstudie 7.3 untersucht. Auf die Mächtigkeit des Tailorings in HERMES sind wir in Abschnitt 3.2.3 eingegangen.

In Ansätzen der dritten Kategorie finden sich Empfehlungen, wann und unter was für Umständen Teile der Vorgehensweise in einem Projekt Anwendung finden sollten und wann nicht. Beispielsweise wird für die Wahl einer Vorgehensweise aus der Crystal Familie eine Matrix mit drei Kriterien vorgeschlagen (vergleiche Abschnitt 2.4.2). Da aus der Crystal Familie bisher nur zwei Vertreter vollständig ausformuliert sind (Crystal Clear und Crystal Orange), lässt sich die Tragfähigkeit dieses Ansatzes kaum überprüfen. Auch für Extreme Programming (XP) gibt es mit RDP [MKRS⁺08] einen Vorschlag für die projektspezifische Auswahl der XP Praktiken. Der Vorschlag sieht die Durchführung eines strukturierten Workshops vor, bei welchem die Projektmitglieder über die im Projekt zur Anwendung kommenden Praktiken entscheiden. Derartige Techniken können dazu beitragen, das Tailoring agiler Vorgehensweisen transparenter und nachvollziehbar zu machen. Aufgrund ihrer informellen Beschreibung und der Tatsache, dass sie in erster Linie *gelebt* und nicht *dokumentiert* werden, bleibt die Nachvollziehbarkeit und Vergleichbarkeit projektspezifischer agiler Vorgehensweisen und ihrer Anpassung ein Thema, welches weiterführende Untersuchungen rechtfertigen würde.

Im Hinblick auf Nachvollziehbarkeit und Vergleichbarkeit der durch Tailoring entstehenden projektspezifischen Vorgehensmodellausprägungen, und um die einfache Durchführung zu ermöglichen, sollte das Tailoring eines Vorgehensmodells klaren Regeln folgen. Zudem sollten die das Tailoring beeinflussenden Projektcharakteristika direkt mit den betroffenen Inhalten des Vorgehensmodells verknüpft sein. Ob das Konfigurationsmodell und das Vorgehensmodell dabei als separate Einheiten oder als *ein Modell* aufgefasst werden, ist schlussendlich unerheblich, so lange beide Modelle über klar definierte Tailoring-Operationen miteinander in Verbindung stehen. Um dem Entwickler eines Vorgehensmodells einen Werkzeugkasten zur Beschreibung komplexer Projektszenarien zu geben, sollten Zusammenhänge zwischen Tailoringkriterien – etwa gegenseitiger Ausschluss – darstellbar sein.

8.2 Lösung und Ergebnisse

Basierend auf den Analysen in den Kapiteln 2, 3 und 4 haben wir in dieser Arbeit das *feature-basierte Tailoring* entwickelt. Dieses eignet sich für alle Vorgehensmo-

8.2 Lösung und Ergebnisse

delle, denen ein Metamodell zugrunde liegt und die aufgrund ihres Umfangs und der Menge der möglichen Variationen ein formal beschriebenes Tailoring-Modell rechtfertigen. Im Folgenden rekapitulieren wir die zentralen Ergebnisse dieser Dissertation.

Konfigurationsmodell

Das feature-basierte Tailoring wurde in dieser Arbeit unabhängig von einem konkreten (Meta-)Modell entwickelt. Hierzu haben wir in Kapitel 4 aus der Analyse populärer Metamodelle ein einfaches Meta-Metamodell für Vorgehensmodelle kondensiert. Da das Meta-Metamodell für Vorgehensmodelle einen Graph beschreibt (mit den Vorgehensmodellinhalten als Knoten und den Beziehungen zwischen diesen Inhalten als Kanten), lässt sich die Semantik des Tailorings beschreiben als die Manipulation eines Graphen.

In Kapitel 5 haben wir ausgehend von möglichen Fragen zur Charakterisierung eines Projekts ein Konfigurationsmodell entwickelt, welches sich wesentlich auf die Konzepte der Feature-Oriented Domain Analysis abstützt. Das Konfigurationsmodell spannt alle Vorgehensmodellvarianten auf, die durch das Tailoring erstellt werden können. Durch die Abstützung auf die Konzepte der Feature-Oriented Domain Analysis ermöglicht das Konfigurationsmodell dem Prozessingenieur die Beschreibung feingranularer Kriterien, ihrer hierarchische Organisation und von Abhängigkeiten zwischen Kriterien. Gleichzeitig lassen sich die Verfahren der Feature-Oriented Domain Analysis nutzen, beispielsweise zur Sicherstellung einer im Sinne des Konfigurationsmodells gültigen Ausprägung des Vorgehensmodells.

Tailoring-Operationen

Die Tailoring-Operationen definieren, wie sich die im Konfigurationsmodell beschriebenen Projektcharakteristika auf die Inhalte des Vorgehensmodells auswirken. Sie stellen die Verbindung zwischen den Inhalten des Vorgehensmodells – in dieser Arbeit abstrakt beschrieben durch das einfache Meta-Metamodell für Vorgehensmodelle – und dem Konfigurationsmodell dar. Ausgehend von einer Analyse möglicher Tailoring-Operationen haben wir in Kapitel 5 eine eindeutige Semantik für Tailoring-Operationen definiert. Durch die klare Semantik der Operationen wird die Wirkung der Situationsfaktoren eines Projekts auf das Vorgehensmodell transparent und nachvollziehbar.

Methode zur Anwendung des feature-basierten Tailorings

Zur Implementierung des feature-basierten Tailorings in einem konkreten Vorgehensmodell haben wir in Kapitel 6 eine Methode zur Anwendung beschrieben. Zur Umsetzung des feature-basierten Tailorings in einem Vorgehensmodell haben wir folgende Anwendungsfälle identifiziert: *Bedarf analysieren*, *Variabilitätspunkte vereinbaren*, *Kriterien vereinbaren*, *Inhalte vereinbaren* und *Werkzeuge implementieren*. Die Methode der Durchführung des Tailorings basierend auf dem hier vorgestellten Ansatz hängt wesentlich von den bei der Implementierung getroffenen Entscheidungen ab. Insbesondere wird das Werkzeug zum Tailoring dessen Durchführung maßgeblich leiten.

Kriterienkatalog

Um die Umsetzung des feature-basierten Tailorings zu unterstützen und insbesondere für die Ausgestaltung des Konfigurationsmodells haben wir in Anhang A einen Kriterienkatalog für das Tailoring von Vorgehensmodellen zusammengetragen. Der Katalog umfasst alle Kriterien, welche in der Literatur für das Tailoring von Vorgehensmodellen als relevant erachtet werden. Da hierbei auch informelle Beschreibungen von Situationsfaktoren für Projekte betrachtet wurden, sind nicht alle Kriterien im Katalog gleichermaßen gut geeignet für die Verwendung im feature-basierten Tailoring. Der Katalog ist zu verstehen als Ausgangsbasis für die Entwicklung eines auf die Organisation, ihre Projekte und das Vorgehensmodell zugeschnittenen Konfigurationsmodells.

Fallstudien

In den Fallstudien in Kapitel 7 haben wir anhand des SE Book (siehe Abschnitt 7.2) und des V-Modell XT (siehe Abschnitt 7.3) die praktische Umsetzung des feature-basierten Tailorings demonstriert. Neben der Demonstration der Machbarkeit sind die Fallstudien selbst wesentliche Ergebnisse, da sie jeweils eine konkrete Implementierung des Ansatzes darstellen. Bei der Fallstudie zum SE Book wurde ein vollständiges, industriell eingesetztes Werkzeug zur Durchführung des feature-basierten Tailorings entwickelt. Mit der für das SE Book angepassten und erweiterten Variante des Process Development Environment wurde zudem ein Werkzeug entwickelt, welches die Erstellung des Konfigurationsmodells durch den Prozessingenieur unterstützt und vereinfacht. Die Ergebnisse der beiden Experimente zur Gegenüberstellung der Tailoring-Ansätze im V-Modell XT, im Eclipse Process Framework (EPF) und im SE Book (als Vertreter des feature-basierten Tailorings) bestätigen und untermauern die in der vorliegenden Arbeit postulierten positiven Eigenschaften des feature-basierten Tailorings.

8.3 Ausblick und offene Forschungsfragen

Das feature-basierte Tailoring legt den Grundstein für die Entwicklung flexibel an die Projektsituation anpassbarer Vorgehensmodelle. Die Strukturen für das Metamodell, die Tailoring-Operationen, die Methode zur Implementierung des feature-basierten Tailorings und der Kriterienkatalog geben einem Prozessingenieur die nötigen Hilfsmittel an die Hand, mächtige Tailoring-Mechanismen im Vorgehensmodell zu verankern. In diesem Abschnitt gehen wir ein auf offene Fragen und Themenbereiche, die anknüpfend an diese Arbeit erforscht werden sollten.

Tailoring zur Projektlaufzeit

Ein Vorgehensmodell lebt während der gesamten Laufzeit des Projekts und muss gegebenenfalls „feinjustiert“ werden, um die Realität des Projekts und seiner Umgebung abzubilden. Insbesondere sind zu Projektbeginn nicht alle Informationen verfügbar, um das Vorgehen über die gesamte Laufzeit des Projekts belastbar festzulegen. Sich ändernde Geschäftsziele, volatile Anforderungen, realisierungstechnische Hürden und vieles mehr können zur Laufzeit des Projekts gravierende Auswirkungen auf das Vorgehen im Projekt haben.

8.3 Ausblick und offene Forschungsfragen

Das in dieser Arbeit entwickelte feature-basierte Tailoring basiert in Teilen auf der idealisierten Annahme, dass die zur Anpassung der Vorgehensweise nötigen Informationen über das Projekt vollständig zu Beginn desselben bekannt sind.

Prinzipiell spricht nichts dagegen, das Tailoring nicht nur einmal zu Beginn, sondern mehrfach während der Laufzeit des Projekts, durchzuführen. Das Ergebnis wäre eine sich mit der Realität des Projekts ändernde Vorgehensweise. Problematisch wird dieses Vorgehen, wenn die durch das Tailoring des Vorgehensmodells entstehenden Artefakte nicht „read-only“ sind. Ein klassisches Beispiel hierfür sind vom Vorgehensmodell zur Verfügung gestellte Vorlagen für zu erstellende Arbeitsergebnisse. Wird so eine Vorlage während des Projekts mit Leben gefüllt, kann sie nicht zu einem späteren Zeitpunkt aufgrund veränderter Projektcharakteristika angepasst werden – zumindest nicht automatisiert. Ebenfalls problematisch wird das Tailoring zur Laufzeit, wenn die Änderung der Projektcharakteristika eine Änderung der Projektvergangenheit nach sich ziehen müsste. Würde beispielsweise nach der Hälfte der Projektdauer festgestellt, dass der Entwicklungsgegenstand erhöhten Sicherheitsanforderungen unterliegt, hätten nach der reinen Lehre des Vorgehensmodells möglicherweise Anforderungsdokumente, welche bereits abgeschlossen sind, zusätzliche und veränderte Inhalte, um die erhöhten Sicherheitsanforderungen zu reflektieren.

Ein möglicherweise gangbarer Weg, die Vorteile des feature-basierten Tailorings zu nutzen und gleichzeitig die Dynamik zur Projektlaufzeit anzuerkennen, könnte sein, das Tailoring *stufenweise* durchzuführen. Ergebnis des Tailorings wären immer nur jene Inhalte des Vorgehensmodells, die für die nächste Projektstufe (etwa bis zum nächsten Meilenstein) benötigt würden. Im Rahmen der vorliegenden Arbeit bleibt ein derartiges Vorgehen eine theoretische Möglichkeit, ohne dass Erfahrungen dazu gesammelt wurden. Insbesondere bliebe auch bei diesem Ansatz zu klären, wie mit einem Tailoring umzugehen ist, welches nach der Lehre des Vorgehensmodells „die Vergangenheit des Projekts“ ändern würde.

Tailoring von agilen Vorgehensweisen

Die in Abschnitt 8.3 diskutierten Fragestellungen zum Tailoring während der Laufzeit des Projekts ergeben sich bei agilen Vorgehensweisen nicht, da hier die kontinuierliche Reflexion und Neuausrichtung der Vorgehensweise wesentlicher Bestandteil der Vorgehensweise selbst ist. Möglich wird dies durch den Umstand, dass agile Vorgehensweisen ihre Manifestation im Wesentlichen in der Kultur des Projektteams finden. Im Vergleich zu reichhaltigen Vorgehensweisen gibt es weniger (Prozess-)Dokumentation, die von einer Neuausrichtung betroffen wäre. Das geringe Maß an Dokumentation lässt vermuten, dass die positiven Eigenschaften des feature-basierten Tailorings, wie Transparenz, Nachvollziehbarkeit und Vergleichbarkeit der Ergebnisse der Anpassung agiler Vorgehensweisen in der Regel nicht zugeschrieben werden können.

Es wäre zu untersuchen, wie, mit welcher Systematik und nach welchen Kriterien in der Praxis die Anpassung agiler Vorgehensweisen vorgenommen wird und ob sich nicht durch eine Systematisierung¹ – beispielsweise durch Orientierung an einem Kriterienkatalog – positive Effekte erzielen ließen.

1 Über die Vorschläge von [MKRS⁺08] hinausgehend.

Integration von Methodenbibliotheken

Im Abschnitt 2.2.2 haben wir diskutiert, dass eine wesentliche Aufgabe des Tailoring darin besteht, das Vorgehensmodell von denjenigen Inhalten zu befreien, die für die Projektsituation keine Bedeutung haben. Die zweite wesentliche Aufgabe des Tailoring besteht in der *Anreicherung* des Vorgehensmodells um spezifische, auf die Projektsituation zutreffende Inhalte. In Abschnitt 2.2.2 haben wir diesen Aspekt der Anpassung *Spezialisierung in der Tiefe* genannt. Die Spezialisierung in der Tiefe lässt sich realisieren, indem *Methodenbausteine* beim Tailoring mit berücksichtigt und je nach Projektcharakteristika in die projektspezifische Ausprägung des Vorgehensmodells aufgenommen werden.

In Abschnitt 6.3.6 haben wir mögliche Optionen zur Aufnahme von Methodenbausteinen in das Tailoring eines Vorgehensmodells diskutiert. Dabei haben wir uns abgestützt auf die in [Ham08] entwickelten Konzepte. Diese Diskussion blieb im Rahmen der vorliegenden Arbeit eine theoretische. Die Kombination der Ansätze aus [Ham08] und der vorliegenden Arbeit und eine kritische Auseinandersetzung damit bleibt Gegenstand zukünftiger Arbeiten.

Diffusion

Mit den Fallstudien zum SE Book in Abschnitt 7.2 und zum V-Modell XT in Abschnitt 7.3 haben wir gezeigt, wie das feature-basierte Tailoring in einem Vorgehensmodell implementiert werden kann.

Neben dem mit diesen Studien erbrachten Nachweis der Machbarkeit hat die Studie zum SE Book in Form des Laborexperiments erste Hinweise auf die Verständlichkeit und Bedienbarkeit des zum SE Book entwickelten Tailoring-Werkzeugs gegeben. Die Studie zum V-Modell XT hat mit dem fiktiven Beispiel zur Evaluierung angedeutet, wie die Grenzen des Tailoring im V-Modell XT mit Hilfe des feature-basierten Tailoring ausgedehnt werden können.

Für das in Abschnitt 1.5 gefasste Ziel des Transfers der Ergebnisse in die Praxis stellt die vorliegende Arbeit einen Anfang und eine Ausgangsbasis dar. Mit zukünftigen Arbeiten sollten weitere Erfahrungen mit dem feature-basierten Tailoring gesammelt werden und die Umsetzbarkeit in der Praxis existierender Anforderungen an das Tailoring nachgewiesen werden. So existieren noch nicht zu allen Tailoring-Kriterien im Kriterienkatalog im Anhang A Erfahrungen oder Beispiele aus der Praxis, welche Wirkung auf die Inhalte eines Vorgehensmodells diese haben sollten. Der Katalog ist aus diesem Grund unter anderem als Aufruf zu weiteren Untersuchungen zu verstehen.

Kriterienkatalog für das Tailoring von Vorgehensmodellen

Im folgenden Katalog sind Tailoring-Kriterien aus der Literatur und aus bekannten Vorgehensmodellen zusammengetragen. Zu jedem Kriterium ist die Quelle und mögliche Auswirkungen auf das Vorgehensmodell angegeben. Der Katalog basiert wesentlich auf den Ergebnissen der in Abschnitt 2.4.4 beschriebenen Untersuchung.

Der Katalog erhebt keinen Anspruch auf Vollständigkeit. Da jede Organisation einmalig ist und jeweils ein einmaliges Projektportfolio hat, sind auch die Kriterien, die für die projektspezifische Anpassung zu wählen sind, jeweils individuell. Der Katalog sollte aber Anhaltspunkte dafür geben, welche Kriterien in der Literatur als wichtig erachtet werden und welche in realen Vorgehensmodellen zum Einsatz kommen. Er stellt damit eine Grundlage dar für die Entwicklung eines eigenen Kriterienkatalogs.

Die hier aufgelisteten Kriterien sind nicht alle gleichermaßen gut geeignet für die Verwendung im feature-basierten Tailoring. Faktoren, die eine Verwendung erschweren können, sind insbesondere eine erschwerte Einschätzung und damit Wertbelegung des Kriteriums und eine unklare Wirkung der Wertbelegung eines Kriteriums auf die Inhalte des Vorgehensmodells. Der Katalog ist als wertungsfreie Zusammenstellung der in der Literatur und in bekannten Vorgehensmodellen gefundenen Kriterien zu verstehen.

A.1 Team

Teamgröße

Die Teamgröße ist ein Indikator für den Kommunikationsaufwand im Projekt. Während 4-6 Projektbeteiligte möglicherweise in einem Raum zusammenarbeiten und kaum Hürden für Kommunikation haben, geht eine größere Anzahl an Teammitgliedern in der Regel mit einem Bedarf für formalisierte Dokumentation und Meetings einher.

Die Teamgröße wird in der Literatur als eines der zentralen Kriterien bei der Wahl einer Vorgehensweise und für deren projektspezifische Ausgestaltung angesehen [XR08, Coc06, CO08, HLW95]. Vereinfacht gilt dabei: je größer das

A.1 Team

Team, umso *schwerer* muss das Vorgehensmodell sein (im Sinne des Maßes aus Abschnitt 3.1).

Agile Vorgehensweisen werden in der Regel nur für kleine Teams empfohlen: für Extreme Programming wird die maximale Teamgröße üblicherweise mit 10 angegeben [BA04, ASRW02]. Crystal Clear ist für Teams zwischen 2 und 6 Personen empfohlen [Coc04] und Crystal Orange eignet sich für Teams mit 20-40 Mitarbeitern [Coc06].

Quelle: [XR08, Coc06, CO08, HLW95, MFWL⁺12], SE Book

Wirkung: Je größer die Anzahl an Projektbeteiligten, umso größer in der Regel der Bedarf für formalisierte Dokumentation und Meetings. Derartige Vorgaben wiederum sind wesentlicher Bestandteil jedes Vorgehensmodells.

Abbildung im feature-basierten Tailoring: Durch Definition geeigneter Kategorien (*klein, mittel, groß*) oder durch Wertebereiche (*0-10, 10-50, 50-100, usw.*).

Örtliche Verteilung

Analog zur → **Teamgröße** hat die örtliche Verteilung des Teams Einfluss auf die Intensität und Art der Kommunikation im Projekt. Ist das Team über mehrere Sprachräume und Zeitzonen verteilt, wird informelle Kommunikation („Flurfunk“) erschwert, wenn nicht unmöglich. Studien legen sogar nahe, dass ab einer Entfernung von 50 Metern weitere Entfernung zwischen Teammitgliedern für die Art und Intensität der Kommunikation keinen Unterschied mehr macht [All84]. Demzufolge würde bereits ein Team, welches auf zwei Gebäude verteilt ist, als geographisch verteilt gelten.

Fast alle agile Vorgehensweisen sind für die Anwendung in einem Team an einem Standort empfohlen. Für Crystal Clear lautet die Empfehlung sogar ausdrücklich, dass das Team zusammen in einem Raum arbeitet [Coc04].

Quelle: [KK08, MFWL⁺12], SE Book

Wirkung: Wie beim Kriterium der → **Teamgröße** hat die geographische Verteilung des Teams Einfluss auf die Art der im Projekt zu erstellenden Dokumentation und der Besprechungen.

Abbildung im feature-basierten Tailoring: im einfachsten Fall als Ja-/Nein-Schalter. Wenn es Abstufungen gibt, sollten diese möglicherweise ebenfalls abgebildet werden – beispielsweise *Verteiltes Team mit einer Sprache* und *Verteiltes Team mit mehreren Sprachen*.

Fluktuation im Team

Verlässt ein Mitglied das Team, führt der Weggang unweigerlich zu einem Verlust am kollektiven Wissen. Um das Wissen dennoch zu konservieren, sollte das Vorgehensmodell entsprechende Gegenmaßnahmen vorsehen.

Quelle: [XR08]

Wirkung: Eine hohe Fluktuation im Team sollte im Vorgehensmodell zu erhöhten Anforderungen hinsichtlich der Dokumentation im Projekt führen.

Abbildung im feature-basierten Tailoring: beispielsweise durch einen Gradmesser in der Art *niedrig, durchschnittlich, hoch*.

Haben die Teammitglieder in früheren Projekten bereits zusammengearbeitet?

Kennen sich die Teammitglieder bereits aus früheren Projekten und sind mit den jeweiligen Fähigkeitsprofilen vertraut, besteht ein geringerer Bedarf an formalisierter Kommunikation.

Quelle: [XR08]

Wirkung: Kennen sich die Teammitglieder bereits aus früheren Projekten, können die Vorgaben hinsichtlich der formalisierten Dokumentation und für formale Meetings im Projekt reduziert werden.

Abbildung im feature-basierten Tailoring: im einfachsten Fall durch einen Ja-/Nein-Schalter.

Wie gut arbeiten die Mitglieder des Teams zusammen?

Dieses Kriterium steht in enger Beziehung zu dem vorherigen (siehe → **frühere Zusammenarbeit des Teams**). Arbeiten die Teammitglieder gut und eng zusammen, ist davon auszugehen, dass die Kommunikation funktioniert und Abläufe eingespielt sind. Insgesamt resultiert daraus ein geringerer Bedarf an formalisierter Kommunikation.

Quelle: [XR08]

Wirkung: Die Vorgaben in Bezug auf Dokumentation und für formalisierte Besprechungen können reduziert werden, sofern nicht andere Faktoren, z. B. hohe Anforderungen an → **Safety und Security**, dagegen sprechen.

Abbildung im feature-basierten Tailoring: im einfachsten Fall durch einen Ja-/Nein-Schalter (das Team arbeitet gut zusammen oder das Team arbeitet nicht gut zusammen bzw. die Abläufe im Team sind noch nicht eingespielt). Alternativ durch Definition einer geeigneten Skala.

Vertrautheit des Teams mit der Domäne

Die Vertrautheit des Projektteams mit der Domäne des Entwicklungsgegenstands kann maßgeblichen Einfluss auf das Projektrisiko und auf den Projekterfolg haben.

Hat das Team Projekte in einem vergleichbaren Umfeld schon öfter durchgeführt und ist mit den Fragestellungen der Domäne vertraut, ist der Projekterfolg wahrscheinlicher als wenn sich die Teammitglieder mit der Materie erst vertraut machen müssen. Ist die Domäne nicht gut verstanden, steigt die Wahrscheinlichkeit, „am Problem vorbei zu entwickeln“. Zudem werden Realisierungsrisiken aufgrund des mangelnden Verständnisses möglicherweise nicht oder spät erkannt.

Quelle: [XR08, HLW95, MFWL⁺12]

Wirkung: Ist die Vertrautheit des Teams mit der Domäne gering, sollte das Vorgehensmodell Vorgaben zum Aufbau des Domänenverständnisses beinhalten. Beitragen kann hierzu enger Kontakt mit dem → **Kunden** und den → **Endanwendern**. Weiterhin ist besonderes Gewicht auf die Anforderungserhebung zu legen.

Abbildung im feature-basierten Tailoring: z. B. durch eine XOR-Verknüpfung von *nicht vertraut, einige Kenntnisse, sehr gutes Verständnis*.

Vertrautheit des Teams mit den Werkzeugen

Geringe Vertrautheit des Teams mit den Werkzeugen stellt ein Risiko für das Projekt dar.

Quelle: [XR08]

Wirkung: Falls die Vertrautheit mit den Werkzeugen gering ist, könnte das Vorgehensmodell Maßnahmen zur Weiterbildung des Teams vorsehen. Weiterhin sollten Methoden zur Verbreitung des Wissens innerhalb des Teams genutzt werden. Die für den Wissensaufbau erforderliche Zeit sollte in der Projektplanung berücksichtigt werden.

Abbildung im feature-basierten Tailoring: beispielsweise durch eine XOR-Verknüpfung von *nicht vertraut, einige Kenntnisse, sehr gutes Verständnis*.

Vertrautheit des Teams mit der verwendeten Technologie

Wie auch in Bezug auf die → **Domäne** und die verwendeten → **Werkzeuge** stellt eine geringe Kenntnis der verwendeten Technologie ein Risiko für das Projekt dar.

Quelle: [XR08]

Wirkung: Die Vertrautheit des Teams mit der verwendeten Technologie muss aufgebaut werden. Hierzu eignen sich möglicherweise Schulungen (die in der Planung berücksichtigt werden sollten) und die Entwicklung von → **Prototypen**.

Abbildung im feature-basierten Tailoring: beispielsweise durch eine XOR-Verknüpfung von *nicht vertraut, einige Kenntnisse, sehr gutes Verständnis*.

Vertrautheit des Teams mit dem Vorgehensmodell

Das Vorgehensmodell sollte dem Team Hinweise geben bei Fragestellungen wie: Was sind meine Verantwortlichkeiten? An wen kann ich mich wenden? Was, wo und wie muss ich dokumentieren? Wo stehen wir? Was steht als nächstes an?

Nicht nur formale Aspekte des Vorgehensmodells, auch die darin enthaltenen Methoden erfordern Vertrautheit im Umgang und Anwendung. Je nach Hintergrund der Teammitglieder müssen zur erfolgreichen Anwendung agiler Praktiken, wie etwa *Pair-Programming* oder *geteilter Code*, zunächst psychologische Barrieren überwunden werden.

Quelle: [XR08, HLW95]

Wirkung: Im Tailoring könnte es eine Forderung sein, dass bei der Durchführung komplexer (siehe → **Komplexität**) oder innovativer Projekte (siehe → **Innovationsgrad**) oder bei der Entwicklung eines Systems mit hohen Anforderungen an → **Safety und Security** eine hohe Vertrautheit des Teams mit dem Vorgehensmodell besteht.

Abbildung im feature-basierten Tailoring: beispielsweise durch eine XOR-Verknüpfung von *nicht vertraut, einige Kenntnisse, sehr gutes Verständnis*.

A.2 Inneres Umfeld

Prototypenentwicklung

Im V-Modell XT wird durch das Projektmerkmal *Prototypenentwicklung* festgelegt, ob die Systementwicklung im Projekt einer prototypischen Entwicklungsstrategie folgen soll.

Bei diesem Kriterium gibt es zwei bemerkenswerte Aspekte:

- Dieses Projektmerkmal wählt im V-Modell XT keine statischen Inhalte, wie Artefakte oder Aktivitäten, sondern eine zeitliche Anordnung für Meilensteine während der Projektdurchführung. Abhängig vom konkreten Vorgehensmodell, der Organisation und dem Team ist die Wahl einer Strategie zur Durchführung eines Projekts ein interessantes Thema für die projektspezifische Anpassung.
- Die prototypische Systementwicklung entspricht im Wesentlichen einer agilen Vorgehensweise. Es läge nahe, agile Praktiken und Vorgaben – beispielsweise zur Einbindung eines Kunden vor Ort (*On Site Customer*) – in die Dokumentation des Vorgehensmodells aufzunehmen (vergleiche auch Abschnitt 4.4.1 und → **Verfügbarkeit des Kunden**).

Quelle: [RHB⁺06]

Wirkung: Verallgemeinert vom konkreten Kriterium der Prototypenentwicklung: wenn das Vorgehensmodell Variabilität bei den Durchführungsstrategien erlaubt, sollte durch ein derartiges Kriterium eine entsprechende Durchführungsstrategie gewählt werden. Das Kriterium Prototypenentwicklung im V-Modell XT bedeutet konkret die Wahl einer agilen Durchführungsstrategie. Damit einhergehen sollten entsprechende agile Praktiken und Vorgaben in der Dokumentation des Vorgehensmodells.

Abbildung im feature-basierten Tailoring: durch einen einfachen Ja-/Nein-Schalter.

A.2.1 Management

Klarheit und Eindeutigkeit des Projektvorschlags

In [RHB⁺06] wird der Zweck des Projektvorschlags beschrieben als „die systematische Darstellung der Informationen und Daten, die deutlich machen, dass die Durchführung eines Projektes notwendig, rentabel und nutzbringend ist“.

Der Projektvorschlag dient insbesondere der Kommunikation der Projektziele nach innen ins Projekt und nach außen, beispielsweise in Richtung der Geschäftsleitung (und nimmt damit unter Umständen Einfluss auf die → **Unterstützung der Geschäftsleitung**).

Ein klar und eindeutig formulierter Projektvorschlag kann die Priorisierung von Anforderungen erleichtern, indem diese leichter hinsichtlich ihres Beitrags zu den übergeordneten Projektzielen gemessen werden können.

Quelle: [XR08, HLW95]

Wirkung: Laut [XR08] sollte ein unklar formulierter Projektvorschlag zu einer stärkeren Gewichtung der Anforderungserhebung im Projekt führen. Dies sollte vom Vorgehensmodell reflektiert werden, beispielsweise durch Vorgaben für formalisierte Anforderungswshops und hinsichtlich der Dokumentation von Anforderungen.

A.2 Inneres Umfeld

Abbildung im feature-basierten Tailoring: Sollte dieses Kriterium beim Tailoring berücksichtigt werden, ist es realistischerweise am besten durch einen Ja-/Nein-Schalter darstellbar. Eine abgestufte Einschätzung der Klarheit des Projektvorschlags erscheint unrealistisch – eine abgestufte Wirkung auf Inhalte des Vorgehensmodells erst recht.

Verfügbarkeit der Geschäftsleitung

Die Verfügbarkeit der Geschäftsleitung ist vor allem wichtig in Krisen des Projekts. Probleme – beispielsweise mit dem Kunden – die nicht im Projekt selbst gelöst werden können, müssen zur Geschäftsleitung eskaliert werden. Auch bei Projektfortschrittsentscheidungen ist die Geschäftsleitung involviert.

Schlechter Kontakt zur Geschäftsleitung kann Verzögerungen ins Projekt bringen und dazu führen, dass die Ergebnisse des Projekts sich nicht mit den übergeordneten Geschäftszielen decken (siehe auch → **Unterstützung der Geschäftsleitung**).

Quelle: [XR08]

Wirkung: Wenn kein enger Kontakt zur Geschäftsleitung besteht, sollte wenigstens eine gute → **Verfügbarkeit des Kunden** gewährleistet sein.

Besonders wichtig kann die Verfügbarkeit der Geschäftsleitung sein in Projekten, die ein knappes → **Projektbudget** oder eine knapp bemessene → **Laufzeit** haben, da in solchen Situationen Krisen vorprogrammiert sind.

Abbildung im feature-basierten Tailoring: durch einfachen Ja-/Nein-Schalter oder durch abgestufte Einschätzung, etwa *Reagiert stark verzögert, Reagiert mit gelegentlichen Verzögerungen, Reagiert prompt.*

Unterstützung der Geschäftsleitung

Die Unterstützung der Geschäftsleitung ist entscheidend zur erfolgreichen Durchführung des Projekts, insbesondere in Projekten, in denen aufgrund eines knappen → **Projektbudgets**, einer knappen → **Laufzeit**, hoher → **Komplexität** oder einem nicht → **eingespielten Team** Krisen vorprogrammiert sind.

Auch sollte in Projekten, in denen der → **Kunde** schlecht verfügbar ist, wenigstens die Geschäftsleitung hinter dem Vorhaben stehen.

Die Unterstützung kann gleich zu Beginn des Projekts gesichert werden, beispielsweise durch einen → **klaren und eindeutigen Projektvorschlag**.

Die Unterstützung der Geschäftsleitung kann unter Umständen gefördert werden durch die → **Erhebung quantitativer Kennzahlen**. Diese machen den Status und Fortschritt des Projekts für die Geschäftsleitung transparent und leichter zu beurteilen.

Quelle: [XR08, HLW95]

Wirkung: Bei geringer Unterstützung der Geschäftsleitung sollte wenigstens die → **Verfügbarkeit des Kunden** gewährleistet sein. Der Unterstützung zuträglich sein können die → **Erhebung quantitativer Kennzahlen** und die Einbeziehung der Geschäftsleitung, beispielsweise durch regelmäßige Termine und Demonstration der Projektergebnisse.

Abbildung im feature-basierten Tailoring: etwa durch *hohe Unterstützung, geringe Unterstützung.*

Projektbudget

Die finanzielle Ausstattung des Projekts beeinflusst den Detaillierungsgrad und die Formalität des Vorgehensmodells.

Quelle: [XR08]

Wirkung: Bei einem knappen Projektbudget kann unter Umständen die Reduktion des Detaillierungsgrades der Projektdokumentation und des „Zeremoniells“ im Projekt (vergleiche Abschnitt 3.1) sinnvoll sein [XR08]. Gleichzeitig sollte der Einsatz der finanziellen Mittel überwacht werden, beispielsweise durch besondere Vorgaben zum → **kaufmännischen Projektmanagement**.

Abbildung im feature-basierten Tailoring: beispielsweise durch eine Einteilung *ausreichend* und *kritisch*.

Projektlaufzeit

Die Projektlaufzeit als absolutes Maß aber auch im Verhältnis zum Umfang des Projektgegenstands kann Einfluss auf das Vorgehensmodell haben.

Als absolutes Maß bedeutet eine längere Projektlaufzeit in der Regel höheres Projektrisiko. Gründe hierfür sind: → **Mitarbeiterfluktuation** über einen längeren Zeitraum ist wahrscheinlicher, sich änderndes Marktumfeld (man entwickelt an den Bedürfnissen vorbei oder wird von der Konkurrenz überholt), sich änderndes inneres Umfeld (wechselnde Geschäftsleitung und fehlende → **Unterstützung derselben**), volatile Anforderungen.

Quelle: [XR08, MFWL⁺¹²]

Wirkung: Im Verhältnis zum Umfang des Projektgegenstands kann eine knappe Projektlaufzeit ähnliche Wirkung wie ein knappes → **Projektbudget** haben und beispielsweise ein Vorgehen mit Timeboxes nahelegen.

Abbildung im feature-basierten Tailoring: durch eine diskretisierte Skala, etwa eine XOR-Verknüpfung von *0-3 Monate*, *3-6 Monate*, *6-12 Monate*, *12-24 Monate*, *länger*.

Projekt-/Servicetyp

Der Projekt-/Servicetyp beschreibt die Art des Projekts und damit wesentlich die Art des Vorgehensmodells.

Wenn z. B. ein → **Altsystem** zu erweitern, abzulösen oder zu warten ist, so muss dieses analysiert werden. Handelt es sich hingegen um ein reines Beratungsprojekt, gibt es keine Systementwicklung und die Vorgaben für Systemtests müssen durch andere Maßnahmen zur Qualitätssicherung der Projektergebnisse ersetzt werden. Wieder in großen Teilen unterschiedliche Vorgaben gelten für ein Projekt, bei dem es um den Test eines bestehenden Systems geht.

Quelle: SE Book, [RHB⁺⁰⁶]

Wirkung: Der Projekt-/Servicetyp nimmt Einfluss auf weite Teile des Vorgehensmodells. Wie ist die Anforderungsanalyse durchzuführen? Ist ein → **Altsystem** zu analysieren? Gibt es eine Entwicklungsphase? Spielen Einführung/Rollout und Betrieb eine Rolle? Welche Iterationen sind zu planen und woraus besteht eine Iteration?

Der Projekt-/Servicetyp gibt unter Umständen die → **fachliche** und → **technische** Art der Lösung vor, beispielsweise wenn es sich um ein Projekt zur Wartung, Weiterentwicklung oder Migration handelt.

A.2 Inneres Umfeld

Abbildung im feature-basierten Tailoring: beispielsweise durch eine XOR-Verknüpfung von *Neuentwicklung, Wartung, Erweiterung, Migration, Consulting, Service/Betrieb, Einführung, Test*

Beauftragung von Sublieferanten

Werden im Rahmen des Projekts Teile der Entwicklungsaufgaben an Sublieferanten ausgelagert, sind entsprechende Anforderungsdokumente zu erstellen (bzw. aus den Gesamtanforderungen zu extrahieren), Angebote einzuholen, Lieferanten auszuwählen, die Beauftragung durchzuführen, der Projektfortschritt des oder der Sublieferanten zu überwachen und entsprechende Abnahmen durchzuführen.

Quelle: SE Book, [RHB⁺06]

Wirkung: Das Vorgehensmodell sollte entsprechende Artefakte und Aktivitäten und geeignete Schnittstellen im Prozess zur Synchronisierung mit dem Sublieferanten beinhalten.

Abbildung im feature-basierten Tailoring: durch einfaches Ja-/Nein-Kriterium. Gibt es eine festgelegte Menge der in Frage kommenden Sublieferanten, eventuell eine Auswahl der entsprechenden Lieferanten, etwa eine OR-Verknüpfung von *Lieferant A, Lieferant B, Lieferant C*.

Handelt es sich um ein Auftraggeber- oder um ein Auftragnehmerprojekt?

Im V-Modell XT wird durch den *Projekttyp* üblicherweise unterschieden, ob es sich bei dem vorliegenden Projekt um ein Auftraggeber- oder ein Auftragnehmerprojekt handelt. Handelt es sich nicht um ein Auftragsprojekt – beispielsweise bei der Entwicklung eines hausinternen Systems oder bei Produktentwicklung – so nimmt das Projekt oder die Organisation beide Rollen ein.

In [RHB⁺06] heißt es zu den unterschiedlichen Projektrollen: „In der Projektrolle AG/AN wird genau ein V-Modell-Projekt durchgeführt, um ein System oder ein organisationsspezifisches Vorgehensmodell selbst zu entwickeln. In der Projektrolle AG wird die Systemerstellung auf Basis von festgelegten Anforderungen an einen oder mehrere Auftragnehmer vergeben. In der Projektrolle AN wird ein Systementwicklungsprojekt auf Basis von vom Auftraggeber festgelegten Anforderungen durchgeführt.“

Quelle: [RHB⁺06]

Wirkung: Je nach Projektrolle sind im Projekt unterschiedliche Ergebnisse zu erstellen und mit dem Auftraggeber bzw. Auftragnehmer auszutauschen.

Abbildung im feature-basierten Tailoring: durch ein Kriterium mit den XOR-verknüpften Werten *Auftraggeberprojekt, Auftragnehmerprojekt, Auftraggeber-/Auftragnehmerprojekt*.

Kaufmännisches Projektmanagement

In der Dokumentation [RHB⁺06] zum entsprechenden Projektmerkmal des V-Modell XT heißt es: „Die kaufmännische Projektplanung und -verfolgung umfasst die Kostenplanung des Projekts und die entsprechende Projektsteuerung. Dies ist insbesondere bei hohen zu erwartenden Kosten wichtig, um den Erfolg eines Projekts zu gewährleisten“.

Hohe zu erwartende Kosten können dabei auch bedeuten, dass das → **Projektbudget** im Verhältnis zum Gesamtaufwand im Projekt knapp ist.

Quelle: [RHB⁺06]

Wirkung: Entsprechende Artefakte und Aktivitäten sollten vom Vorgehensmodell vorgesehen werden und sind einzuplanen. Kaufmännische Aspekte sind bei der Projektsteuerung, z. B. bei der Iterationsplanung, zu berücksichtigen.

Abbildung im feature-basierten Tailoring: durch einfaches Ja-/Nein-Kriterium.

Erhebung quantitativer Kennzahlen

Messungen und Metriken zur Ermittlung quantitativer Projektkennzahlen können dazu beitragen, den Fortschritt und Zustand des Projekts transparent und vergleichbar zu machen. Laut [RHB⁺06] ist dies „z.B. wichtig für die Bewertung der Effektivität eines Entwicklungsprozesses“. Zudem können quantitative Kennzahlen Hinweise auf Fehlentwicklungen im Projekt und für die Projektsteuerung liefern. Sie können zudem förderlich für die → **Unterstützung der Geschäftsleitung** sein, da sie eine Einschätzung des Projektzustands auf objektiver Basis erlauben.

Quelle: [RHB⁺06]

Wirkung: Spielen quantitative Projektkennzahlen eine Rolle, sind geeignete Messungen und Metriken zu vereinbaren und die Kennzahlen im Verlauf des Projekts regelmäßig zu erheben [Eck06]. Das Vorgehensmodell sollte Vorgaben zur Definition der Metriken zu Projektbeginn und zur regelmäßigen Erhebung (beispielsweise bei Erreichen von Meilensteinen im Projekt) beinhalten. Weiterhin ist die Erhebung und Darstellung der Indikatoren bei der Planung der technischen Infrastruktur für das Projekt zu berücksichtigen und gegebenenfalls *Cockpits* und *Dashboards* einzubinden [GGD08, Rei01].

Abbildung im feature-basierten Tailoring: durch einfaches Ja-/Nein-Kriterium.

A.2.2 Technische Umgebung

Verfügbarkeit technischer Unterstützung

Insbesondere bei Projekten, in denen eine neuartige oder vom Team bisher nicht durchdrungene Technologie – beispielsweise in Form von → **Entwicklungssprachen**, → **Datenbanksystemen**, → **Betriebssystemen**, → **Fertigkomponenten**, → **verwendeten Werkzeugen** usw. – zum Einsatz kommt, kann die Verfügbarkeit technischer Unterstützung wesentlich zur Minderung des Projektrisikos beitragen (siehe auch Vertrautheit des Teams mit → **den Werkzeugen** und → **der verwendeten Technologie**).

Die Verfügbarkeit technischer Unterstützung kann sich sowohl auf externe Stellen – beispielsweise den Hersteller eines Betriebssystems, Datenbanksystems oder von Fertigkomponenten – als auch auf organisationsinterne Rollen – etwa die Systemverwaltung – beziehen.

Quelle: [XR08]

Wirkung: Ist die technische Umgebung neu oder unbekannt und die technische Unterstützung gering, kann das Projektrisiko etwa durch das Vorsehen von → **Prototypenentwicklung** im Vorgehensmodell abgemildert werden. Gleichzeitig sollte die eventuell im Projekt erforderliche Überwindung technischer Hürden

A.2 Inneres Umfeld

durch entsprechend konservative Planung im Vorgehensmodell berücksichtigt werden.

Abbildung im feature-basierten Tailoring: durch einfachen Ja-/Nein-Schalter oder durch abgestufte Einschätzung, etwa *Reagiert stark verzögert, Reagiert mit gelegentlichen Verzögerungen, Reagiert prompt.*

Entwicklungssprache

Die Entwicklungssprache gibt Teile der technischen Umgebung, unter Umständen etwa die zu → **verwendenden Werkzeuge** und die → **Betriebssysteme** vor und beeinflusst unter Umständen durch spezielle Architekturmuster und Idio-me die Systemarchitektur.

Quelle: SE Book

Wirkung: Wird die Entwicklungssprache beim Tailoring berücksichtigt, kann diese Information zur Bereitstellung spezifischer Programmier- und Architek-turrichtlinien verwendet werden. Zudem könnte das Vorgehensmodell – zu-sammen mit der Art der → **technischen Lösung** – vorgefertigte und bewährte Schablonen für die Erstellung bestimmter Lösungstypen beinhalten.

Abbildung im feature-basierten Tailoring: Durch eine OR-Verknüpfung von etwa: *C#, Java, HTML, C++, o.ä.*

Integrierte Fremdprodukte

Wenn für Teile des zu entwickelnden Systems bereits Lösungen auf dem Markt existieren, kann eine Integration geeigneter Komponenten sinnvoll sein, da da-durch die Entwicklungszeit verkürzt, das Entwicklungsrisiko minimiert und von Spezialkenntnissen des Komponentenanbieters profitiert werden kann.

Quelle: SE Book, [RHB⁺06]

Wirkung: Laut [RHB⁺06] erfordert „der Einsatz von Fertigprodukten frühzeitig in der Systementwicklung Maßnahmen zur Erfassung der möglichen System-elemente, die Kandidaten für Fertigprodukte sind. Zudem müssen die hierfür auf dem Markt existierenden Lösungen ermittelt und bewertet werden“.

Abbildung im feature-basierten Tailoring: durch einfaches Ja-/Nein-Kriterium oder durch eine OR-Verknüpfung von z. B. *COTS, Open Source, Sonstige*, wenn die die Vorgaben zur Bewertung und Auswahl der Fertigkomponenten abhän-gig von der Art der Fertigkomponenten sind.

Betriebssysteme

Das Betriebssystem der Zielumgebung für die zu entwickelnde Lösung kann unter Umständen die → **Entwicklungssprache** und die zu verwendenden → **Werkzeuge** vorgeben. Weiterhin schränkt das Betriebssystem die in Frage kom-menden → **Fertigkomponenten** ein.

Umgekehrt ist das zu verwendende Betriebssystem möglicherweise durch die Art der → **technischen Lösung** eingeschränkt.

Quelle: SE Book

Wirkung: Ähnlich wie Programmierrichtlinien zu → **Entwicklungssprachen** kann das Vorgehensmodell Hinweise und Empfehlungen, beispielsweise zur

Gestaltung von → **grafischen Benutzeroberflächen** für die zu betrachtenden Betriebssysteme beinhalten.

Soll das zu entwickelnde System auf mehreren Betriebssystemen lauffähig sein, muss dies bereits beim Entwurf des Systems berücksichtigt werden – beispielsweise durch Abstraktion von Betriebssystemschnittstellen und durch Kapselung der vom Betriebssystem abhängigen Anteile des Systems. Weiterhin erfordert die Multi-Plattform-Entwicklung besondere Vorgaben und Infrastruktur für die Tests des Systems.

Abbildung im feature-basierten Tailoring: Etwa durch eine OR-Verknüpfung von *Windows 7, Windows 8, Windows Azure, Linux, Mac OS*.

Datenbanken

Wenn das zu entwickelnde System eine Datenhaltung benötigt, kann das Vorgehensmodell diesen Aspekt der Systementwicklung durch geeignete Vorgaben und Hinweise unterstützen.

Quelle: SE Book

Wirkung: Spielen Datenbanken bei dem zu entwickelnden System eine Rolle, könnte das Vorgehensmodell Vorgaben für die Auswahl eines geeigneten Datenbanksystems und Vorgaben und Hinweise zur Definition der Datenbankschemata beinhalten. Gleichzeitig sollten für den Test des Systems eine geeignete Infrastruktur und Testdaten vorgesehen werden.

Abbildung im feature-basierten Tailoring: Durch einfachen Ja-/Nein-Schalter oder durch eine XOR-Verknüpfung etwa von *Microsoft SQL Server, PostgreSQL, MySQL, CouchDB, Berkeley DB, Hive*, oder ähnliche.

Werkzeuginfrastruktur

Jeder Themenbereich der Systementwicklung lässt sich durch entsprechende (Software-)Werkzeuge unterstützen.

Abhängig vom Umfang des Projekts und des Gewichts und Formalitätsgrades der Vorgehensweise kommen nicht für alle Phasen *Softwarewerkzeuge* zum Einsatz – Anforderungen und Systemarchitektur werden möglicherweise nur am Whiteboard beschrieben und festgehalten.

In jedem Fall sollte in der Vorgehensweise aber Klarheit darüber herrschen, welche Mittel zur Erarbeitung der Projektergebnisse zum Einsatz kommen.

Quelle: SE Book

Wirkung: Im Tailoring des SE Book lässt sich für jeden Themenbereich der Systementwicklung angeben, ob die zu verwendenden Werkzeuge vorgegeben sind oder im Rahmen des Projekts zu evaluieren, auszuwählen und zu beschaffen sind. Je nach Wahl beim Tailoring enthält das Vorgehensmodell die entsprechenden Vorgaben oder nicht.

Abbildung im feature-basierten Tailoring: Etwa durch OR-Verknüpfung von *Werkzeuge zur Anforderungsanalyse vorgegeben, Werkzeuge zum Systementwurf vorgegeben, Werkzeuge zur IDE vorgegeben, Werkzeuge zur Integration vorgegeben, Werkzeuge zum Konfigurationsmanagement vorgegeben, Werkzeuge zur Kommunikation vorgegeben, Werkzeuge zum Problem-/Änderungsmanagement vorgegeben, Werkzeuge zur Qualitätssicherung vorgegeben*.

A.3 Äußeres Umfeld

Juristische Konsequenzen

Je nach Domäne und Branche können aus dem Projekt juristische Konsequenzen für die Organisation erwachsen. Erstens kann eine Verfehlung der Projektziele, etwa hinsichtlich Projektdauer, -kosten und der Qualität des Projektergebnisses, Regressforderungen nach sich ziehen. Zweitens kann ein Fehlverhalten des entwickelten Systems schwerwiegende Folgen, im Extremfall für Leib und Leben von Menschen, nach sich ziehen.

Quelle: [Coc06]

Wirkung: Die Schwere der drohenden juristischen Konsequenzen kann den Grad der Formalität und das Gewicht der Vorgehensweise beeinflussen. Das kann beispielsweise einhergehen mit detaillierten Vorgaben zur Sicherstellung von → **Safety und Security** des zu entwickelnden Systems.

Abbildung im feature-basierten Tailoring: durch einfaches Ja-/Nein-Kriterium oder beispielsweise durch eine abgestufte Angabe des drohenden finanziellen Schadens für die Organisation, etwa *gering, substanziell, existenzbedrohend*.

Anzahl der Interessensgruppen am Projekt

Je größer die Anzahl der Interessensgruppen am Projekt, umso mehr Abstimmungsbedarf und -aufwand gibt es. Eine hohe Anzahl von Interessensgruppen kann sich verzögernd auf den Projektfortschritt auswirken, da beispielsweise zur Priorisierung von Anforderungen und zu Projektfortschrittsentscheidungen alle Interessenten einzubeziehen und eventuell viele divergierende Meinungen zu vereinbaren sind.

Quelle: [XR08]

Wirkung: Der Abstimmungsbedarf mit den Interessensgruppen ist bei der Projektplanung zu berücksichtigen.

Analog zum Kriterium → **Verfügbarkeit des Kunden** sollten die Vorgaben des Vorgehensmodells bei einer hohen Anzahl von Interessensgruppen zur Vermeidung von Missverständnissen Vorgaben zur regelmäßigen Einbeziehung der Interessensgruppen vorsehen.

Abbildung im feature-basierten Tailoring: abhängig von der differenzierten Wirkung auf die Inhalte des Vorgehensmodells beispielsweise *normal* und *hoch*.

Verfügbarkeit von Stakeholdern

Die Verfügbarkeit von Stakeholdern kann als übergeordnetes Kriterium über → **Verfügbarkeit der Geschäftsleitung**, → **Verfügbarkeit des Kunden**, → **Verfügbarkeit von Endanwendern** und sonstigen interessierten Parteien aufgefasst werden.

Quelle: [XR08, MFWL⁺12]

Wirkung: Dieses Kriterium kann sinnvoll sein, wenn beim Tailoring nicht zwischen den verschiedenen Interessensgruppen unterschieden werden soll, beispielsweise weil bei der Wirkung auf die projektspezifische Ausprägung des Vorgehensmodells nicht differenziert wird (bzw. werden kann).

Die Wirkung dieses Kriteriums auf die projektspezifische Ausprägung des Vorgehensmodells entspricht im Wesentlichen der bei den Kriterien → **Verfügbarkeit der Geschäftsleitung**, → **Verfügbarkeit des Kunden** und → **Verfügbarkeit von Endanwendern** beschriebenen.

Abbildung im feature-basierten Tailoring: etwa durch *Alle Stakeholder sind stets greifbar, Stakeholder sind nicht gut greifbar*.

Hintergrund und Wissensstand der Stakeholder

Betreten die Beteiligten mit dem Projekt Neuland, beispielsweise weil der → **fachliche** oder → **technische** Lösungstyp, die → **verwendeten Werkzeuge**, die → **Programmiersprachen**, → **integrierte Fremdprodukte** oder die Vorgehensweise neu sind oder weil das Projekt einen hohen → **Innovationsgrad** oder eine hohe → **Komplexität** hat, sollte dem daraus resultierenden Projektrisiko durch das Vorgehensmodell Rechnung getragen werden.

Dieses Kriterium kann als übergeordnetes Kriterium über den Kriterien → **Vertrautheit des Teams mit der Domäne**, → **Vertrautheit des Teams mit den Werkzeugen**, → **Vertrautheit des Teams mit der verwendeten Technologie**, → **Vertrautheit des Teams mit dem Vorgehensmodell**, → **Vertrautheit der Endanwender mit der Domäne und dem System** und → **Verfügbarkeit von Altsystemen und Dokumenten** aufgefasst werden.

Quelle: [XR08, MFWL⁺12]

Wirkung: Das Vorgehensmodell und die Projektplanung sollten die für den Wissensaufbau erforderlichen Maßnahmen berücksichtigen. Unter Umständen ist ein „Herantasten“ an den Projektgegenstand – beispielsweise durch → **Prototypenentwicklung** – sinnvoll.

Abbildung im feature-basierten Tailoring: etwa durch eine XOR-Verknüpfung von *hohes Wissen und Erfahrung, hohes Wissen aber Einarbeitung in Teilbereichen nötig, komplettes Neuland*.

A.3.1 Kunde

Stabilität der Anforderungen

Die Stabilität der Anforderungen bestimmt maßgeblich die im Projekt zu verfolgende Strategie. Umso instabiler die Anforderungen, desto wichtiger ist eine stete Einbeziehung des Kunden in den Fortschritt des Projekts und damit die → **Verfügbarkeit des Kunden**.

Quelle: [XR08]

Wirkung: Je instabiler die Anforderungen an das zu entwickelnde System sind, umso flexibler muss das Vorgehen im Projekt sein. Bei instabilen Anforderungen liegt ein Vorgehen mit kurzen Iterationen und häufigen Zwischenlieferungen – etwa in Form von → **Prototypen** – nahe.

Das aus instabilen Anforderungen resultierende Projektrisiko sollte durch Gestaltung eines entsprechenden → **Vertrags** zwischen Auftraggeber und Auftragnehmer geteilt werden.

Abbildung im feature-basierten Tailoring: etwa durch XOR-Verknüpfung von *stabil, instabil*.

Kunden-/Produktvorgehensmodell

Der Kunde/Auftraggeber folgt in der Regel eigenen Vorgaben für die Durchführung seiner Projekte, welche die Vorgaben des Auftragnehmers tangieren können.

Quelle: SE Book

Wirkung: Folgt der Kunde etwa dem V-Modell XT, sind die entsprechenden Schnittstellenprodukte der Auftraggeber-/Auftragnehmerschnittstelle des V-Modell XT zu berücksichtigen.

Im Allgemeinen hat das Kunden-/Produktvorgehensmodell Einfluss auf Inhalt und Form der mit dem Kunden auszutauschenden Artefakte.

Weiterhin kann das Kundenvorgehensmodell Einfluss auf den Grad an Zeremoniell im Vorgehen des Projekts haben, beispielsweise, wenn vom Kunden ein bestimmtes CMMI-Level gefordert wird.

Abbildung im feature-basierten Tailoring: durch eine XOR-Verknüpfung der in Frage kommenden Vorgehensmodelle.

Verfügbarkeit des Kunden

Die Verfügbarkeit eines *On Site Customer* trägt dazu bei, dass das zu entwickelnde System den Erwartungen des Kunden entspricht. In einigen agilen Vorgehensweisen, wie z. B. Extreme Programming, wird die Verfügbarkeit des Kunden (*On Site Customer*) sogar gefordert.

Ist der Kunde gleichzeitig Anwender des Systems, so fällt dieses Kriterium zusammen mit der → **Verfügbarkeit von Endanwendern**.

Steht der Kunde im Projekt nicht zur Verfügung, kann er in die Entwicklung einbezogen werden, indem er durch häufige Zwischenlieferungen am Fortschritt teilnimmt. Weiterhin sollte eine höhere Wahrscheinlichkeit für Missverständnisse einkalkuliert werden [JAH00].

Quelle: [XR08, MFWL⁺12]

Wirkung: Steht der Kunde im Projekt nicht zur Verfügung, können häufige Zwischenlieferungen und regelmäßige Telefonkonferenzen und persönliche Treffen zur Vermeidung und Ausräumung von Missverständnissen sinnvoll sein. Dies sollte im Vorgehensmodell entsprechend berücksichtigt werden.

Abbildung im feature-basierten Tailoring: durch einfaches Ja-/Nein-Kriterium.

Vertragsart

Bei der Vertragsart kann grob zwischen Werkvertrag und Dienstvertrag unterschieden werden. Die Art des Vertrags bestimmt entscheidend das Vorgehen im Projekt.

Quelle: SE Book

Wirkung: Die Vertragsart hat Einfluss auf die Anfangsphasen des Projekts und auf die Art und Häufigkeit der Lieferungen an den Kunden.

Auch die Art und Weise der Abnahmen durch den Kunden wird durch die Vertragsart beeinflusst. Erfolgt eine formale Abnahme zum Ende des Projekts oder ist der Kunde (beispielsweise ein *Kunde vor Ort*, siehe → **Verfügbarkeit des Kunden**) kontinuierlich in den Projektfortschritt involviert?

Weiterhin kann die Art des Vertrags Einfluss nehmen auf die Anzahl und Frequenz von Entwicklungszyklen. Werden beispielsweise regelmäßige Zwischenlieferungen vereinbart, sind entsprechende Iterationszyklen einzuplanen. Das Vorgehen im Projekt wird in so einem Fall kaum einem reinen Wasserfallmodell folgen.

Abbildung im feature-basierten Tailoring: XOR-verknüpfte Features *Werkvertrag, Dienstvertrag*.

A.3.2 Endanwender

Verfügbarkeit von Endanwendern

Stehen Endanwender des zu entwickelnden Systems im Projekt zur Verfügung, können und sollten diese bei der Anforderungserhebung unterstützen und über den gesamten Projektverlauf in die Entwicklung einbezogen werden. Wenn der Kunde gleichzeitig Endanwender ist, fällt dieses Kriterium zusammen mit → **Verfügbarkeit des Kunden**.

Quelle: [BA04, HLW95, Coc04]

Wirkung: Steht ein Endanwender im Projekt zur Verfügung, sollte dieser Umstand bei den Vorgaben zur Anforderungserhebung berücksichtigt werden. Die Bedürfnisse und die Meinung des Endanwenders sind einzuholen und zu dokumentieren.

Abbildung im feature-basierten Tailoring: durch einfachen Ja-/Nein-Schalter.

Vertrautheit der Endanwender mit der Domäne und dem System

Hier ist zu unterscheiden zwischen Endanwendern, die im Projekt zur Verfügung stehen und (potenziellen) Endanwendern, deren Bedürfnisse und Meinung im Projekt nicht *direkt* berücksichtigt werden können.

Im ersten Fall handelt es sich um einen *On Site Customer* im Sinne von → **Verfügbarkeit des Kunden** oder → **Verfügbarkeit von Endanwendern**. Seine Erfahrung mit dem System und/oder der Domäne beeinflussen den Beitrag, den er zu den Anforderungen an das System zu leisten imstande ist. Die Erfahrung des *On Site Customer* ist schwer einzuschätzen – erst recht vor bzw. zu Projektbeginn.

Im zweiten Fall handelt es sich allgemein um (potenzielle) Endanwender. Geht es bei dem zu entwickelnden System beispielsweise um ein völlig neues Produkt am Markt, können die Endanwender noch keine Erfahrung mit dem System selbst und/oder mit vergleichbaren Produkten (z. B. der Konkurrenz) gesammelt haben. In diesem Fall besteht zum System einerseits erhöhter Erklärungsbedarf und es ist ein besonderes Gewicht auf die Erhebung der Anforderungen zu legen.

Zur Illustration folgende Beispiele: bei der Entwicklung des ersten iPhone kann angenommen werden, dass die Endanwender geringe Vertrautheit mit dem zu entwickelnden System hatten, da zum damaligen Zeitpunkt weder eine Vorversion noch vergleichbare Produkte existierten. Wird im Projekt hingegen der Online-Auftritt einer Bank entwickelt, kann davon ausgegangen werden, dass die Endanwender eine Vorstellung des Systems und der zu erwartenden Funktionen haben.

Quelle: [XR08, HLW95, MFWL⁺12]

A.4 Projektgegenstand

Wirkung: Erstens kann eine geringe Vertrautheit der Endanwender mit dem System den Bedarf für → **Schulungen** begründen. Zweitens kann die Vertrautheit der Endanwender mit dem System den Aufwand beeinflussen, der in die Anforderungserhebung zu investieren ist. Sind die Anwender gut mit dem System vertraut, können sie ihre Wünsche formulieren. Sind sie nicht gut vertraut, wissen sie vielleicht gar nicht, was sie wollen. „If I'd asked my customers what they wanted, they'd have said a faster horse.“¹

Abbildung im feature-basierten Tailoring: beispielsweise durch eine XOR-Verknüpfung von *keine Vertrautheit*, *geringe Vertrautheit*, *gute Vertrautheit*, *hohe Vertrautheit*.

Schulungen erforderlich

Sind im Projekt Schulungen für die Endanwender des zu entwickelnden Systems durchzuführen, ist dies im Vorgehensmodell zu berücksichtigen.

Quelle: SE Book

Wirkung: Das Vorgehensmodell enthält Artefakte für Schulungsunterlagen und Aktivitäten zur Erstellung derselben.

Abbildung im feature-basierten Tailoring: In der Regel dürfte ein einfacher Ja-/Nein-Schalter ausreichen.

A.4 Projektgegenstand

Komplexität

Die Komplexität des Projektgegenstands setzt sich zusammen aus der Komplexität der Anwendungsdomäne, der umzusetzenden Geschäftsprozesse und der Technik des Systems selbst.

Quelle: [HLW95]

Wirkung: Je komplexer das zu entwickelnde System ist, umso „schwerer“ (im Sinne des Gewichtsmaßes aus Abschnitt 3.1) sollte das Vorgehensmodell sein. Die Komplexität kann unter Umständen beherrscht werden, indem eine *Divide and Conquer*-Strategie verfolgt wird: die Aufgaben werden in kleine, beherrschbare „Häppchen“ unterteilt. Insbesondere eignet sich zum Nachweis der Umsetzbarkeit möglicherweise die → **Prototypenentwicklung**.

Abbildung im feature-basierten Tailoring: möglicherweise durch abgestuftes Kriterium, etwa *gering*, *normal*, *hoch*.

Innovationsgrad

Der Innovationsgrad des Projekts hängt ab von dem Innovationsgrad des Projektgegenstands selbst, dem Innovationsgrad des Entwicklungsansatzes (siehe auch → **Vertrautheit des Teams mit dem Vorgehensmodell**) und der Aktualität der eingesetzten technischen Mittel, wie z. B. → **Programmiersprachen**, → **Werkzeuge**, → **Betriebssystem**, usw.

Quelle: [HLW95, MFWL⁺12]

¹ Dieser berühmte Satz wird Henry Ford zugeschrieben, ohne dass sich hierfür eine belastbare Quelle angeben ließe [Woo12]

Wirkung: Ein hoher Innovationsgrad kann ein Risiko für das Projekt darstellen. Er kann unter Umständen eine → **Prototypenentwicklung** nahelegen. Bei fachlicher Innovation sollte insbesondere die → **Unterstützung der Geschäftsleitung** sichergestellt sein. Beitragen hierzu kann eine klare und eindeutige Beschreibung der Projektziele im → **Projektvorschlag**.

Abbildung im feature-basierten Tailoring: möglicherweise durch abgestuftes Kriterium, etwa *gering, normal, hoch*.

Altsystem

Dieses Kriterium fällt unter Umständen zusammen mit dem → **Projekt-/Service-typ**. Wird der Projekt-/Servicetyp im Tailoring nicht berücksichtigt, kann es sinnvoll sein, wenigstens zu unterscheiden, ob ein Altsystem in die Betrachtung einbezogen werden muss oder nicht.

Quelle: [RHB⁺06]

Wirkung: Ein Altsystem ist bei der Anforderungserhebung besonders zu berücksichtigen und schränkt den Gestaltungsspielraum in allen anderen Phasen des Projekts maßgeblich ein. Beispielsweise gibt es die → **Entwicklungssprache**, die zu verwendende → **Werkzeugumgebung** und die technische Infrastruktur, z. B. in Form des → **Betriebssystems**, vor.

Soll das System sowohl technisch als auch fachlich überarbeitet werden, ist in der Regel eine Migration durchzuführen [RHB⁺06]. Dementsprechend sollte das Vorgehensmodell ein Migrationskonzept und entsprechende Aktivitäten zur Erstellung desselben beinhalten.

Abbildung im feature-basierten Tailoring: durch einfachen Ja-/Nein-Schalter. Ja: Das Projekt befasst sich mit der Weiterentwicklung und/oder Migration eines Altsystems. Nein: Altsysteme sind in diesem Projekt kein Betrachtungsgegenstand.

Verfügbarkeit von Altsystemen oder Dokumenten

Spielt im Projekt ein → **Altsystem** eine Rolle, ist ein wesentlicher Faktor für den Erfolg des Projekts die Verfügbarkeit von Dokumentation zum Altsystem und die Möglichkeit zum Einblick in das Altsystem.

Quelle: [XR08, MFWL⁺12]

Wirkung: Die Verfügbarkeit von Dokumentation zum Altsystem und die Möglichkeit, Einblick in das Altsystem zu nehmen, bestimmt den Aufwand, der in die Analyse zu investieren ist. Dies könnte und sollte im Vorgehensmodell berücksichtigt werden.

Abbildung im feature-basierten Tailoring: möglicherweise durch ein abgestuftes Kriterium, etwa *keine Dokumentation vorhanden, Dokumentation vorhanden, aber eingehende Analyse erforderlich* und *umfassende Dokumentation vorhanden*.

Branche

Die Branche ist auf jeden Fall eine wichtige Information als Controlling-Größe im Projektportfolio.

In Bezug auf ein projektspezifisches Vorgehensmodell kann die Branche Hinweise geben auf die einzuhaltenden Gesetze, Normen und Vorschriften. Auch

A.4 Projektgegenstand

können je nach Branche Regressforderungen, und damit → **Juristische Konsequenzen** drohen, oder – z. B. bei Entwicklung von Systemen im medizinischen oder militärischen Bereich – besondere Anforderungen hinsichtlich → **Safety und Security** gelten.

Weiterhin kann die Branche Rückschlüsse zulassen über die → **Domänenkenntnis der Projektbeteiligten**.

Quelle: SE Book

Wirkung: Spielt die Branche des Projektgegenstands bei der projektspezifischen Anpassung des Vorgehensmodells eine Rolle, wird sich dies am wahrscheinlichsten in der Aufnahme branchenspezifischer Regularien in das Vorgehensmodell niederschlagen.

Abbildung im feature-basierten Tailoring: z. B. eine XOR-Verknüpfung von *Automotive, Public, Finance, Health, Telecommunications*.

Lösungstyp fachlich

Die Art der fachlichen Lösung kann dazu genutzt werden, im Vorgehensmodell Best Practices, Architekturmuster, usw. zur Entwicklung entsprechender Systeme aufzunehmen. Zudem kann die Art der fachlichen Lösung Einfluss haben auf die Art der → **technischen Lösung** und damit auf technische Charakteristika, wie Verwendung von → **Datenbanksystemen**, → **Programmiersprachen**, → **Werkzeuginfrastruktur**, usw.

Quelle: SE Book

Wirkung: Das Vorgehensmodell könnte spezifische Vorgaben, Richtlinien und Hinweise zur Erstellung eines entsprechenden Systems beinhalten.

Abbildung im feature-basierten Tailoring: beispielsweise durch eine OR-Verknüpfung von *Management Information System, Business Intelligence, Billing, Supply Chain Management, Product Lifecycle Management, Customer Relationship-Management*, usw.

Lösungstyp technisch

Die Art der technischen Lösung kann die Verwendung spezifischer Best-Practices und Architekturmuster nahelegen. Zudem nimmt die Art der technischen Lösung Einfluss auf andere Kriterien: Eine Webanwendung und eine Dialoganwendung haben immer eine → **Benutzeroberfläche**, womit die entsprechenden Vorgehensmodellinhalte Teil der projektspezifischen Ausprägung sein sollten. Bei einer Webanwendung kommen möglicherweise nur bestimmte → **Betriebssysteme** in Frage und es ist davon auszugehen, dass eine → **Datenbank** eine Rolle spielt.

Quelle: SE Book

Wirkung: Abhängig von der Art der technischen Lösung sollte das Vorgehensmodell Best-Practices, Hinweise zur Verwendung von Architekturmustern, usw. beinhalten. Zudem wird die Art der technischen Lösung möglicherweise die Auswahlmöglichkeiten anderer (technischer) Kriterien einschränken (siehe oben).

Abbildung im feature-basierten Tailoring: beispielsweise durch eine XOR-Verknüpfung von *Systemsoftware, Batchsystem, Webanwendung, Dialoganwendung*

Safety-/Security

Die Ansprüche hinsichtlich Safety und Security des Projektgegenstands beeinflussen das Maß an „Ceremony“ im Vorgehensmodell. Sicherheitsanforderungen sind schon beim Entwurf des Systems zu berücksichtigen, Ergebnisse im Projekt sind besonders zu prüfen und zu dokumentieren und auf Schwachstellen zu testen. Bei sehr hohen Anforderungen kann auch ein formaler Nachweis der Korrektheit des entwickelten Systems erforderlich oder sinnvoll sein.

Methoden zur Entwicklung von Systemen mit hohen Anforderungen an Security finden sich beispielsweise in: [FGH⁺10, HH09, SHH11].

Methoden zur Entwicklung Safety-kritischer Systeme sind z. B.: [HH09, HS97, HH05, Hei95].

Die Nichterfüllung von Anforderungen an Safety und Security des zu entwickelnden Systems kann möglicherweise → **Juristische Konsequenzen** nach sich ziehen.

Quelle: SE Book, [RHB⁺06, Coc06]

Wirkung: Die Ansprüche hinsichtlich Safety und Security beeinflussen maßgeblich den Grad an Formalität des Vorgehensmodells. Erhöhte Anforderungen erfordern konstruktive Sicherstellung der Sicherheit des Systems und striktes Vorgehen bei der Prüfung und Dokumentation der Projektergebnisse.

Abbildung im feature-basierten Tailoring: durch einfachen Ja-/Nein-Schalter oder z. B. durch XOR-Verknüpfung von *normale Ansprüche*, *erhöhte Ansprüche*, *hohe Ansprüche*, *extrem hohe Ansprüche*.

Nicht-funktionale Anforderungen

Nicht-funktionale Anforderungen an das zu entwickelnde System können einen großen Einfluss auf das zu wählende Vorgehen im Projekt nehmen. Ist das Laufzeitverhalten des Systems beispielsweise ein entscheidender Erfolgsfaktor, müssen Verfahren zur effizienten Berechnung entwickelt und ihr Verhalten erprobt werden, bevor diese in die Entwicklung einfließen können.

Ähnlich wie gesteigerte Ansprüche an → **Safety und Security** können sich nicht-funktionale Anforderungen im kompletten Entwicklungsprozess – von der Systemarchitektur über die Entwicklung bis zu den Tests – niederschlagen.

Quelle: Eigene Überlegungen

Wirkung: Nicht-funktionale Anforderungen können sich in allen Bereichen der Systementwicklung niederschlagen und beispielsweise Vorgaben und Hinweise schon beim Entwurf des Systems beeinflussen. Gleichzeitig kommt möglicherweise der internen Forschung (zur Entwicklung effizienter Verfahren) eine größere Bedeutung zu und → **Prototypenentwicklung** kann zum Nachweis der nicht-funktionalen Eigenschaften des Systems erforderlich sein.

Abbildung im feature-basierten Tailoring: Analog zu → **Domänenkriterien** kann es bei nicht-funktionalen Anforderungen sinnvoll sein, diese direkt als Tailoring-Kriterien vorzusehen.

Werden in den Projekten der Organisation beispielsweise Systeme entwickelt, bei denen das Laufzeitverhalten eine besondere Rolle spielt, läge ein Kriterium *Ansprüche an das Laufzeitverhalten* mit den Werten *normale Ansprüche*, *erhöhte Ansprüche*, *hohe Ansprüche*, *extrem hohe Ansprüche* nahe.

Handelt es sich um ein System mit erhöhten Speicherverbrauch (oder dieser steht auf der zum Einsatz kommenden Hardware nur begrenzt zur Verfügung),

A.4 Projektgegenstand

könnte die Vereinbarung eines entsprechenden Kriteriums, etwa *Ansprüche an das Speicherverhalten* sinnvoll sein.

Hardwareentwicklung

Wird im Projekt zusammen mit Software auch Hardware entwickelt, sind die Hardwareeinheiten zu spezifizieren, die Entwicklung zu planen, die Verteilung der Softwareeinheiten auf Hardwareeinheiten zu spezifizieren, spezielle Integrationstests durchzuführen, usw.

Quelle: [RHB⁺06]

Wirkung: Das Vorgehensmodell sollte entsprechende Artefakte und Aktivitäten beinhalten.

Abbildung im feature-basierten Tailoring: durch einfachen Ja-/Nein-Schalter.

Schnittstellen zu Nachbarsystemen

Wenn das zu entwickelnde System Teil eines größeren Gesamtsystems ist, sind Schnittstellen zu den Nachbarsystemen zu spezifizieren. Dieses Kriterium fällt möglicherweise zusammen mit dem Kriterium → **Systemintegrationstests**.

Quelle: SE Book

Wirkung: Das Vorgehensmodell sollte entsprechende Dokumentation und Aktivitäten beinhalten.

Abbildung im feature-basierten Tailoring: durch einfachen Ja-/Nein-Schalter.

Benutzeroberfläche

Wenn das zu entwickelnde System eine Benutzeroberfläche beinhaltet, sind Entwürfe der Benutzeroberfläche zu erstellen und Richtlinien zur Ergonomie und zum Erscheinungsbild zu beachten. Zudem erfordert ein System mit Benutzeroberfläche besondere Testverfahren.

Quelle: SE Book, [RHB⁺06]

Wirkung: Das Vorgehensmodell sollte die oben skizzierten Richtlinien, Artefakte und Aktivitäten zur Erstellung eines Systems mit Benutzeroberfläche beinhalten.

Abbildung im feature-basierten Tailoring: durch einfachen Ja-/Nein-Schalter.

Systemintegrationstests

Ist das zu entwickelnde System Teil eines größeren Gesamtsystems, sind Systemintegrationstests zu planen, zu spezifizieren und durchzuführen. Dieses Kriterium fällt möglicherweise zusammen mit dem Kriterium → **Schnittstellen zu Nachbarsystemen**.

Quelle: SE Book

Wirkung: Artefakte und entsprechende Aktivitäten für die Planung, Spezifikation und Durchführung von Systemintegrationstests sollten im Vorgehensmodell enthalten sein.

Abbildung im feature-basierten Tailoring: durch einfachen Ja-/Nein-Schalter.

Domänenkriterien

Abhängig von der Entwicklungsdomäne ergeben sich möglicherweise Kriterien, die in das Tailoring des Vorgehensmodells Eingang finden sollten. Ein Beispiel für Domänenkriterien haben wir in der Fallstudie zum SE Book mit den Cloud-Best-Practices gesehen (vergleiche Abschnitt 7.2.3).

Quelle: SE Book, Eigene Überlegungen

Wirkung: Die Wirkung auf die Inhalte einer projektspezifischen Ausprägung des Vorgehensmodells hängen von Art Bedeutung der Domänenkriterien ab.

Abbildung im feature-basierten Tailoring: abhängig von Art und Inhalt der Domänenkriterien.

ANHANG B

Fragebogen zum Evaluierungsexperiment

Auf den folgenden Seiten findet sich der Fragebogen zu dem in Abschnitt 7.4 durchgeführten Experiment zum Vergleich von Tailoring-Ansätzen.

FRAGEBOGEN ZUM VERGLEICH VON TAILORING-ANSÄTZEN

EINLEITUNG

Ziel des Experiments zum Vergleich von Tailoring-Ansätzen ist es, die Stärken und Schwächen der folgenden Tailoring-Ansätze und der dabei zum Einsatz kommenden (Software-)Werkzeuge herauszuarbeiten:

- **Eclipse Process Framework** (mit dem Werkzeug **EPF Composer**)
- **V-Modell XT** (mit dem Werkzeug **V-Modell XT Projektassistent**)
- **SE Book** (mit dem Werkzeug **Process Suite Generator**)

Der grobe **Ablauf des Experiments** ist wie folgt:

- Sie erstellen eine projektspezifisch angepasste Vorgehensmodell-Variante mit einem der drei Tailoring-Ansätze auf Basis einer Projektbeschreibung in Form eines Projektsteckbriefs. Der Projektsteckbrief befindet sich im Anhang.
- Begleitend zur Durchführung des Tailorings beantworten Sie die Fragen im Fragebogen.
- Der detaillierte Ablauf und die einzelnen Schritte sind im Fragebogen an der jeweiligen Stelle beschrieben.

FRAGEBOGEN

1. Mit **welchem Ansatz** führen Sie das Tailoring durch?

2. Lesen Sie nun den Projektsteckbrief durch und beantworten Sie folgende Frage: Wie viele Projektcharakteristika können Sie im Projektsteckbrief identifizieren, die Sie für das Tailoring eines Vorgehensmodells als relevant betrachten? **Markieren Sie die relevanten Passagen** im Projektsteckbrief und **tragen Sie eine Zahl ein**.

3. Führen Sie nun die projektspezifische Anpassung des Vorgehensmodells mit dem entsprechenden Software-Werkzeug durch und **speichern Sie das Ergebnis ab**.

B Fragebogen zum Evaluierungsexperiment

4. Nach der Durchführung des Tailorings beantworten Sie bitte folgende Fragen. Alle Fragen sind auf einer Skala von -2 (starke Ablehnung) über 0 (weder Ablehnung noch Zustimmung) bis +2 (starke Zustimmung) zu beantworten.

- a. Mir war jederzeit klar, **wie ich beim Tailoring vorzugehen** hatte.

Stimme sehr zu	Stimme zu	Weder/noch	Stimme nicht zu	Stimme gar nicht zu

- b. Ich wusste, **welche Entscheidungen** ich zu treffen hatte.

Stimme sehr zu	Stimme zu	Weder/noch	Stimme nicht zu	Stimme gar nicht zu

- c. Mir war jederzeit ersichtlich, **wie sich meine Entscheidungen** auf die Inhalte des Vorgehensmodells **auswirken**.

Stimme sehr zu	Stimme zu	Weder/noch	Stimme nicht zu	Stimme gar nicht zu

5. Denken Sie zurück an Frage 2. Möglicherweise haben Sie während des Tailorings weitere relevante Charakteristika identifiziert. Tragen Sie hier **die aktualisierte Zahl** der identifizierten Projektcharakteristika ein.

6. Wie viele der in Frage 5 identifizierten Projektcharakteristika konnten Sie mit dem Tailoring abbilden? **Tragen Sie eine Zahl ein.**

7. Zuletzt geben Sie ihre Einschätzung zu folgenden beiden Fragen ab:

- a. Das Tailoring ist mir:

Sehr leicht gefallen	Leich gefallen	Weder/noch	Schwer gefallen	Sehr schwer gefallen

- b. Der Tailoring-Ansatz (und das Software-Werkzeug) benötigen:

Kaum Einarbeitung/ Vorkenntnisse	Etwas Einarbeitung/ Vorkenntnisse	Viel Einarbeitung/ Vorkenntnisse	Sehr viel Einarbeitung/Vorkenntnisse

VIELEN DANK FÜR'S MITMACHEN!

GUTSCHEIN



Literaturverzeichnis

- [All84] ALLEN, THOMAS J.: *Managing the Flow of Technology: Technology Transfer and the Dissemination of Technological Information Within the R&D Organization*, Band 1 der Reihe MIT Press Books. The MIT Press, December 1984.
- [ASRW02] ABRAHAMSSON, P., O. SALO, J. RONKAINEN und J. WARSTA: *Agile Software Development Methods - Review and Analysis*. Technischer Bericht 478, VTT PUBLICATIONS, 2002.
- [BA04] BECK, KENT und CYNTHIA ANDRES: *Extreme Programming Explained: Embrace Change (2nd Edition)*. Addison-Wesley Professional, 2004.
- [Bal00] BALZERT, HELMUT: *Lehrbuch der Softwaretechnik Band 1/2*. Spektrum Akademischer Verlag, 2. Auflage, 2000. ISBN 3-8274-0480-0.
- [Bal07] BALDUINO, RICARDO: *Introduction to Eclipse Process Framework: EPF Composer and OpenUP/Basic*. In: *EclipseCon 2007*, 2007.
- [Bat05] BATORY, DON S.: *Feature Models, Grammars, and Propositional Formulas*. In: OBBINK, J. HENK und KLAUS POHL (Herausgeber): *SPLC*, Band 3714 der Reihe *Lecture Notes in Computer Science*, Seiten 7–20. Springer, 2005.
- [BBvB⁺01] BECK, KENT, MIKE BEEDLE, ARIE VAN BENNEKUM, ALISTAIR COCKBURN, WARD CUNNINGHAM, MARTIN FOWLER, JAMES GRENNING, JIM HIGHSMITH, ANDREW HUNT, RON JEFFRIES, JON KERN, BRIAN MARICK, ROBERT C. MARTIN, STEVE MELLOR, KEN SCHWABER, JEFF SUTHERLAND und DAVE THOMAS: *Manifesto for Agile Software Development*. Retrieved May 29, 2007, from <http://www.agilemanifesto.org/>, 2001.
- [Bec03] BECK, K.: *Extreme Programming*. Addison-Wesley, München, 2003. ISBN-13: 978-3827321398.
- [BEJ06] BUSCHERMÖHLE, R., H. ECKHOFF und B. JOSKO: *Success – Erfolgs- und Misserfolgskriterien bei der Durchführung von Hard- und Softwareentwicklungsprojekten in Deutschland*. BIS-Verlag der Carl von Ossietzky Universität Oldenburg, 2006. ISBN 978-3-8142-2035-2.
- [Ben83] BENINGTON, HERBERT D.: *Production of Large Computer Programs*. *Annals of the History of Computing*, 5(4):350–361, Oktober 1983.

- [BFH⁺10] BROY, MANFRED, MARTIN FEILKAS, MARKUS HERRMANNSSDOERFER, STEFANO MERENDA und DANIEL RATTI: *Seamless Model-Based Development: From Isolated Tools to Integrated Model Engineering Environments*. Proceedings of the IEEE, 98(4):526 – 545, apr 2010. available at <http://dx.doi.org/10.1109/JPROC.2009.2037771>.
- [BKPS04] BÖCKLE, G., P. KNAUBER, K. POHL und K. SCHMID: *Software-Produktlinien – Methoden, Einführung und Praxis*. dpunkt.verlag, 2004. ISBN-13: 978-3898642576.
- [BM04] BIRON, PAUL V. und ASHOK MALHOTRA (Herausgeber): *XML Schema Part 2: Datatypes*. W3C Recommendation. W3C, Second Auflage, Oktober 2004.
- [Boe88] BOEHM, BARRY W.: *A Spiral Model of Software Development and Enhancement*. Computer, 21:61–72, May 1988.
- [Boe02] BOEHM, BARRY: *Get Ready for Agile Methods, with Care*. Computer, 35(1):64–69, Januar 2002. doi:10.1109/2.976920.
- [BP10] BERRE, DANIEL LE und ANNE PARRAIN: *The Sat4j library, release 2.2*. JSAT, 7(2-3):59–6, 2010.
- [BR87] BASILI, V. R. und H. D. ROMBACH: *Tailoring the Software Process to Project Goals and Environments*. In: *ICSE '87: Proceedings of the 9th international conference on Software Engineering*, Seiten 345–357, Los Alamitos, CA, USA, 1987. IEEE Computer Society Press.
- [Bri96] BRINKKEMPER, SJAAK: *Method engineering: engineering of information systems development methods and tools*. Information and Software Technology, 38(4):275 – 280, 1996. Method Engineering and Meta-Modelling.
- [Bro87] BROOKS, JR., FREDERICK P.: *No Silver Bullet Essence and Accidents of Software Engineering*. Computer, 20:10–19, April 1987.
- [Bro95] BROOKS, FREDERICK P.: *The mythical man-month : essays on software engineering*. Addison-Wesley Pub. Co, Jubiläums Auflage, August 1995.
- [BSH99] BRINKKEMPER, SJAAK, MOTOSHI SAEKI und FRANK HARMSSEN: *Meta-modelling based assembly techniques for situational method engineering*. Information Systems, 24(3):209 – 228, 1999. <ce:title>10th International Conference on Advanced Information Systems Engineering</ce:title>.
- [Bun10] BUNDESVERWALTUNGSAMT, BUNDESSTELLE FÜR INFORMATIONSTECHNIK, BIT 7: STANDARDS UND METHODEN: *Das V-Modell XT Bund*. Online http://www.bit.bund.de/nn_490662/BIT/DE/Standards__Methoden/V-Modell_20XT/node.html, 2010.
- [CE00] CZARNECKI, KRYSZTOF und ULRICH EISENECKER: *Generative Programming: Methods, Tools, and Applications*. Addison-Wesley Professional, Juni 2000.
- [Chr92] CHROUST, G.: *Modelle der Software-Entwicklung*. R. Oldenbourg Verlag München Wien, 1992. ISBN: 3-486-21878-6.
- [Chr07] CHROUST, G.: *Psychologische Widerstände bei der Einführung computer-gestützter Vorgehensmodelle*. 14. Workshop der Fachgruppe WI-VM 2007 der GI. Garching b. München, Germany, 2007.

- [CKK11] COSTACHE, DANIELA, GEORG KALUS und MARCO KUHRMANN: *Design and Validation of Feature-based Process Model Tailoring - A Sample Implementation of PDE*. In: *Proceedings of the 8th European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEM/FSE 2011)*, Seiten 464–467. ACM Press, Sept. 2011.
- [CO08] COLEMAN, GERRY und RORY O’CONNOR: *Investigating software process in practice: A grounded theory perspective*. *J. Syst. Softw.*, 81(5):772–784, Mai 2008.
- [Coc00a] COCKBURN, ALISTAIR: *Balancing Lightness with Sufficiency*. *Cutter IT Journal*, 13:26–33, November 2000.
- [Coc00b] COCKBURN, ALISTAIR: *Selecting a Project’s Methodology*. *IEEE Softw.*, 17:64–71, July 2000.
- [Coc04] COCKBURN, ALISTAIR: *Crystal Clear: A human-powered methodology for small teams*. Addison-Wesley Professional, Erste Auflage, 2004.
- [Coc06] COCKBURN, A.: *Agile Software Development: The Cooperative Game (agile software development series)*. Addison-Wesley Professional, 2006.
- [CW07] CZARNECKI, KRZYSZTOF und ANDRZEJ WASOWSKI: *Feature Diagrams and Logics: There and Back Again*. In: *Proceedings of the 11th International Software Product Line Conference, SPLC ’07*, Seiten 23–34, Washington, DC, USA, 2007. IEEE Computer Society.
- [Deu87] DEUTSCHER NORMENAUSSCHUSS: *DIN 69901: Projektwirtschaft – Projektmanagement – Begriffe*. In: *Projektmanagement; Projektmanagementsysteme*, 1987.
- [DGG06] DAVID, JEAN-LUC, MICKEY GOUSSET und ERIK GUNVALDSON: *Professional Team Foundation Server*. Wrox Press Ltd., Birmingham, UK, UK, 2006.
- [Eck06] ECKERSON, WAYNE W.: *Performance dashboards: measuring, monitoring, and managing your business*. John Wiley and Sons, 2006.
- [Ecl11a] ECLIPSE FOUNDATION: *Das Eclipse Process Framework (EPF) Online Portal*. Online, <http://www.eclipse.org/epf>, Mai 2011.
- [Ecl11b] ECLIPSE FOUNDATION: *OpenUP auf dem EPF Online Portal*. Online, <http://epf.eclipse.org/wikis/openup/>, Juni 2011.
- [ES03] EÉN, NIKLAS und NIKLAS SÖRENSON: *An Extensible SAT-solver*. In: GIUNCHIGLIA, ENRICO und ARMANDO TACHELLA (Herausgeber): *SAT*, Band 2919 der Reihe *Lecture Notes in Computer Science*, Seiten 502–518. Springer, 2003.
- [FGH⁺10] FABIAN, BENJAMIN, SEDA GÜRSES, MARITTA HEISEL, THOMAS SANTEN und HOLGER SCHMIDT: *A Comparison of Security Requirements Engineering Methods*. *Requirements Engineering – Special Issue on Security Requirements Engineering*, 15(1):7–40, 2010.
- [FK07] FRITZSCHE, M. und P. KEIL: *Kategorisierung etablierter Vorgehensmodelle und ihre Verbreitung in der deutschen Software-Industrie*. Interner Bericht des Projekts IOSEW TUM-I0717, Technische Universität München, 2007.
- [Fra10] FRANK, ULRICH: *Zur methodischen Fundierung der Forschung in der Wirtschaftsinformatik*. In: ÖSTERLE, HUBERT, ROBERT WINTER

- und WALTER BRENNER (Herausgeber): *Gestaltungsorientierte Wirtschaftsinformatik: Ein Plädoyer für Rigor und Relevanz*, Seiten 35–44, St. Gallen, 2010.
- [Fri06] FRIEDRICH, J.: *Technische und semantische Transformation von Vorgehensmodellen*. Diplomarbeit, Technische Universität München, 2006.
- [FRO03] FITZGERALD, BRIAN, NANCY L. RUSSO und TOM O’KANE: *Software development method tailoring at Motorola*. Commun. ACM, 46:64–70, April 2003.
- [GGD08] GLUCHOWSKI, PETER, ROLAND GABRIEL und CARSTEN DITTMAR: *Management Support Systeme und Business Intelligence - Computergestützte Informationssysteme für Fach- und Führungskräfte*. Springer, Berlin u. a., 2. Auflage, 2008.
- [Gna05] GNATZ, M.: *Vom Vorgehensmodell zum Projektplan*. Doktorarbeit, Technische Universität München, 2005.
- [GQ95] GINSBERG, M.P. und L.H. QUINN: *Process tailoring and the software capability maturity model*. Technischer Bericht CMU/SEI-94-TR-024, Software Engineering Institute, Pittsburgh, PA, Nov. 1995.
- [Gre01] GRENNING, JAMES: *Launching Extreme Programming at a Process-Intensive Company*. IEEE Softw., 18:27–33, November 2001.
- [Ham08] HAMMERSCHALL, U.: *Flexible Methodenintegration in Vorgehensmodelle – (to be published)*. Doktorarbeit, Technische Universität München, 2008.
- [Har97] HARMSSEN, A. F.: *Situational Method Engineering*. Doktorarbeit, University of Twente, Utrecht, January 1997.
- [Hei95] HEISEL, MARITTA: *Six Steps Towards Provably Safe Software*. In: *Proceedings of the 14th International Conference on Computer Safety, Reliability and Security (SAFECOMP)*, Seiten 191–205. Springer London, 1995.
- [HH05] HEISEL, MARITTA und DENIS HATEBUR: *A Model-Based Development Process for Embedded Systems*. In: *Proc. Workshop on Model-Based Development of Embedded Systems*. Technical University of Braunschweig, 2005. Number: TUBS-SSE-2005-01, Available at <http://www.sse.cs.tu-bs.de/publications/MBEES-Tagungsband.pdf>.
- [HH09] HATEBUR, DENIS und MARITTA HEISEL: *A Foundation for Requirements Analysis of Dependable Software*. In: *Proceedings of the International Conference on Computer Safety, Reliability and Security (SAFE-COMP)*, LNCS 5775, Seiten 311–325. Springer, 2009.
- [Hig02] HIGHSMITH, JIM: *Agile Software Development Ecosystems*. Addison Wesley, 2002. ISBN 0-201-76043-6.
- [HLW95] HARMSSEN, FRANK, IVO LUBBERS und GERARD WIJERS: *Success-driven Selection of Fragments for Situational Methods: The S3 Model*. In: *Proceedings of the Second International Workshop on Requirements Engineering: Foundations of Software Quality*, Band 13 der Reihe *Aachener Beiträge zur Informatik*, Seiten 104–115, 1995.
- [HS97] HEISEL, MARITTA und CARSTEN SÜHL: *Methodological Support for Formally Specifying Safety-Critical Software*. In: *Proceedings 16th Inter-*

- national Conference on Computer Safety, Reliability and Security (SAFECOMP)*, Seiten 295–308. Springer London, 1997.
- [HSF92] HENDERSON-SELLERS, BRIAN und COLIN FREEMAN: *OLMS - An Object Library Management System*. In: POTTER, JOHN, MARIO TOKORO und BERTRAND MEYER (Herausgeber): *TOOLS (6)*, Seiten 175–180. Prentice Hall, 1992.
- [HSR10] HENDERSON-SELLERS, BRIAN und JOLITA RALYTÉ: *Situational Method Engineering: State-of-the-Art Review*. *J. UCS*, 16(3):424–478, 2010.
- [Inf04] INFORMATIKSTRATEGIEORGAN BUND ISB: *Grundwissen zu HERMES*. BBL, Verkauf Bundespublikationen, CH-3003 Bern, 2004.
- [JAH00] JEFFRIES, RON E., ANN ANDERSON und CHET HENDRICKSON: *Extreme Programming Installed*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2000.
- [KBB⁺09] KITCHENHAM, BARBARA, O PEARL BRERETON, DAVID BUDGEN, MARK TURNER, JOHN BAILEY und STEPHEN LINKMAN: *Systematic literature reviews in software engineering – A systematic literature review*. *Information and Software Technology*, 51(1), 2009.
- [KCH⁺90] KANG, K. C., S. G. COHEN, J. A. HESS, W. E. NOVAK und A. S. PETERSON: *Feature-Oriented Domain Analysis (FODA) Feasibility Study*. Technischer Bericht, Carnegie-Mellon University Software Engineering Institute, November 1990.
- [KK03] KROLL, P. und P. KRUCHTEN: *The Rational Unified Process Made Easy – A Practitioner's Guide to RUP*. Addison-Wesley, 2003.
- [KK08] KUHRMANN, MARCO und GEORG KALUS: *Providing Integrated Development Processes for Distributed Development Environments*. In: *Workshop on Supporting Distributed Team Work at Computer Supported Cooperative Work (CSCW 2008)*, Nov. 2008.
- [KK13] KALUS, GEORG und MARCO KUHRMANN: *Criteria for Software Process Tailoring: A Systematic Review*. In: *Proceedings of International Conference on Software and Systems Process (ICSSP 2013)*, Seiten 171–180. ACM Press, 2013.
- [KKC09] KUHRMANN, MARCO, GEORG KALUS und GERHARD CHROUST: *Tool-Support for Software Development Process*. In: *Social, Managerial, and Organizational Dimensions of Enterprise Information Systems*, Seiten 213–231. IGI Global, 2009. ISBN: 978-1-60556-856-7.
- [KKTW10] KUHRMANN, MARCO, GEORG KALUS, MANUEL THEN und EUGEN WACHTEL: *From Design to Tools: Process Modeling and Enactment with PDE and PET*. In: *Proceedings of Third International Workshop on Academic Software Development Tools and Techniques (WASDeTT-3), co-located with the 25th IEEE/ACM International Conference on Automated Software Engineer*, Sept. 2010.
- [KKWB10] KUHRMANN, MARCO, GEORG KALUS, EUGEN WACHTEL und MANFRED BROY: *Visual Process Model Design using Domain-specific Languages*. In: *Proceedings of SPLASH Workshop on Flexible Modeling Tools 2010*. ACM Press, Okt. 2010.
- [KLS11] KUHRMANN, M., C. LANGE und A. SCHNACKENBURG: *A Survey on the Application of the V-Modell XT in German Government Agencies*. In: *EuroSPI Roskilde TODO*, Jun 2011.

- [Kru11] KRUCHTEN, PHILIPPE: *A plea for lean software process models*. In: *Proceedings of the 2011 International Conference on Software and Systems Process, ICSSP '11*, Seiten 235–236, New York, NY, USA, 2011. ACM.
- [KTF09] KUHRMANN, MARCO, THOMAS TERNITÉ und JAN FRIEDRICH: *Das V-Modell XT anpassen*. Informatik im Fokus. Springer, 2009.
- [KTS⁺09] KASTNER, CHRISTIAN, THOMAS THUM, GUNTER SAAKE, JANET FEIGENSPAN, THOMAS LEICH, FABIAN WIELGORZ und SVEN APEL: *FeatureIDE: A tool framework for feature-oriented software development*. In: *2009 IEEE 31st International Conference on Software Engineering*, Seiten 611–614. IEEE, Mai 2009.
- [Kuh06] KUHRMANN, M.: *Projektspezifische Anpassungen nach dem Tailoring des V-Modell XT durchführen*. In: BISKUP, HUBERT und RALF KNEUPER (Herausgeber): *13. Workshop der Fachgruppe WI-VM der Gesellschaft für Informatik e.V. (GI) zum Thema: Nutzen und Nutzung von Vorgehensmodellen*, Seiten 27–42. Shaker Verlag, mar 2006.
- [Kuh07] KUHRMANN, MARCO: *Konstruktion modularer Vorgehensmodelle*. Doktorarbeit, Technische Universität München, 2007.
- [Lik32] LIKERT, R: *A technique for the measurement of attitudes*. *Archives of Psychology*, 140(140):1–55, 1932.
- [LL04] LINGS, B. und B. LUNDELL: *Method-in-Action and Method-in-Tool: Some Implications for CASE*. In: *Proc. 6th International Conference on Enterprise Information Systems (ICEIS 2004)*, 2004.
- [MBC09] MENDONCA, MARCILIO, MOISES BRANCO und DONALD COWAN: *S.P.L.O.T.: software product lines online tools*. In: *Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications, OOPSLA '09*, Seiten 761–762, New York, NY, USA, 2009. ACM.
- [MF11] MENDEZ-FERNANDEZ, D.: *Requirements Engineering: Artefact-Based Customisation*. Doktorarbeit, Technische Universität München, 2011.
- [MFWL⁺12] MÉNDEZ FERNÁNDEZ, DANIEL, STEFAN WAGNER, KLAUS LOCHMANN, ANDREA BAUMANN und HOLGER DE CARNE: *Field study on requirements engineering: Investigation of artefacts, project parameters, and execution strategies*. *Inf. Softw. Technol.*, 54(2):162–178, Februar 2012.
- [Mic12a] MICROSOFT CORPORATION: *Microsoft Solver Foundation*. Online, März 2012.
- [Mic12b] MICROSOFT CORPORATION: *MSF for Agile Software Development v5.0*. Online, März 2012. Version 5.0.
- [Mic12c] MICROSOFT CORPORATION: *Visual Studio Scrum 1.0*. Online, März 2012. Version 1.0.
- [MKRS⁺08] MIRAKHORLI, MEHDI, ABDORRAHMAN KHANIPOUR RAD, FERREIDON SHAMS, MARYAM PAZOKI und ABBAS MIRAKHORLI: *RDP technique: a practice to customize xp*. In: *Proceedings of the 2008 international workshop on Scrutinizing agile practices or shoot-out at the agile corral, APOS '08*, Seiten 23–32, New York, NY, USA, 2008. ACM.

- [MWC09] MENDONÇA, M., A. WASOWSKI und K. CZARNECKI: *SAT-based analysis of feature models is easy*. In: *Proceedings of the Software Product Line Conference*, 2009.
- [OMG07] OMG: *Unified Modeling Language (UML): Superstructure – Version 2.1.1*. Technischer Bericht, Object Management Group, 2007.
- [OMG08] OMG: *Software Process Engineering Metamodel Specification Version 2.0*. Technischer Bericht, Object Management Group, April 2008.
- [OMG11a] OMG: *Meta Object Facility (MOF) Core Specification –Version 2.4.1*. Meta Object Facility (MOF) Core Specification, Object Management Group, 2011.
- [OMG11b] OMG: *OMG MOF 2 XMI Mapping Specification*. Technischer Bericht, Object Management Group, August 2011.
- [OMG11c] OMG: *Unified Modeling Language (UML): Infrastructure – Version 2.4.1*. Technischer Bericht, Object Management Group, 2011.
- [PF02] PALMER, STEPHEN R. und JOHN M. FELSING: *A Practical Guide to Feature-Driven Development*. Prentice Hall, 1 Auflage, Februar 2002.
- [PJg06] PÄR J. ÅGERFALK, BRIAN FITZGERALD: *Exploring the Concept of Method Rationale: A Conceptual Tool to Understand Method Tailoring*. In: SIAU, KENG (Herausgeber): *Advanced Topics in Database Research*, Band 5, Seiten 63–78. IGI Global, Juni 2006. ISBN: 9781591409359.
- [Por09] PORST, ROLF: *Fragebogen - Ein Arbeitsbuch*. Verlag für Sozialwissenschaften, Wiesbaden, 2. Auflage Auflage, 2009.
- [PPLB07] PEDREIRA, O., M. PIATTINI, M.R. LUACES und N.R. BRISABOA: *A Systematic Review of Software Process Tailoring*. SIGSOFT Software Engineering Notes, 32(3):1–6, 2007.
- [Rei01] REICHMANN, THOMAS: *Controlling mit Kennzahlen und Managementberichten : Grundlagen einer systemgestützten Controlling-Konzeption*. Vahlen, München, 6., überarb. und erw. Aufl. Auflage, 2001.
- [RH09] RUNESON, P. und M. HÖST: *Guidelines for Conducting and Reporting Case Study Research in Software Engineering*. Empirical Software Engineering, 14(2):131–164, 2009.
- [RHB⁺06] RAUSCH, A., R. HÖHN, M. BROY, K. BERGNER und S. HÖPPNER: *Das V-Modell XT*. eXamen.press. Springer, 2006.
- [Rom05] ROMBACH, D.: *Integrated Software Process and Product Lines*. In: LI, MINGSHU, BARRY BOEHM und LEON J. OSTERWEIL (Herausgeber): *Unifying the Software Process Spectrum, International Software Process Workshop, SPW 2005, Beijing, China, May 25-27*, Lecture Notes in Computer Science, Seiten 83–90. Springer, 2005.
- [Roy70] ROYCE, W.: *Managing the Development of Large Software Systems*. In: *Proc. IEEE Wescon*, Seiten 1–9, August 1970.
- [sBFea10] ÖSTERLE, HUBERT, JÖRG BECKER, ULRICH FRANK und ET AL.: *Memorandum zur gestaltungsorientierten Wirtschaftsinformatik*. Zeitschrift für betriebswirtschaftliche Forschung, 11:664–669, 2010.
- [Sch01] SCHUH, P.: *Recovery, redemption, and extreme programming*. Software, IEEE, 18(6):34–41, 2001.
- [Sch04] SCHWABER, K.: *Agile Project Management with Scrum*. Microsoft Press, 2004.

Literaturverzeichnis

- [SEI11] SOFTWARE ENGINEERING INSTITUTE, CARNEGIE MELLON: *Software Product Lines*. Online: <http://www.sei.cmu.edu/productlines>, 2011. Visit: 2011-06-06.
- [SHH11] SCHMIDT, HOLGER, DENIS HATEBUR und MARITTA HEISEL: *Software Engineering for Secure Systems: Academic and Industrial Perspectives*, Seiten 32–74. IGI Global, 2011.
- [Som01] SOMMERVILLE, IAN: *Software Engineering*. Pearson Studium, Juni 2001.
- [SR03] STEPHENS, M. und D. ROSENBERG: *Extreme Programming Refactored: The Case Against XP*. APress, 2003.
- [Sta04] STANDISH GROUP INTERNATIONAL: *Chaos Reports*. Online, 2004. http://www.standishgroup.com/chaos_resources/index.php.
- [Stö11] STÖCKER, HOLGER: *Feingranulares Tailoring des V-Modell XT*. Bachelorarbeit, Technische Universität München, 2011.
- [Str98] STRAHRINGER, SUSANNE: *Ein sprachbasierter Metamodellbegriff und seine Verallgemeinerung durch das Konzept des Metaisierungsprinzips*. In: *Modellierung '98, Proceedings des GI-Workshops in Münster*, 11.-13. März 1998, 1998.
- [Ter09] TERNITÉ, THOMAS: *Process Lines: A Product Line Approach Designed for Process Model Development*. In: *Proceedings of the 2009 35th Euro-micro Conference on Software Engineering and Advanced Applications, SEAA '09*, Seiten 173–180, Washington, DC, USA, 2009. IEEE Computer Society.
- [Ter10] TERNITÉ, THOMAS: *Variability of Development Models*. Doktorarbeit, Clausthal University of Technology, 2010.
- [TK09] TERNITÉ, T. und M. KUHRMANN: *Das V-Modell XT 1.3 Metamodell*. Technischer Bericht TUM-I0905, Technische Universität München, 2009.
- [Tur06] TURNER, M.: *Microsoft Solutions Framework Essentials*. Microsoft Press, 2006. ISBN: 0-7356-2353-8.
- [tV97] TER HOFSTEDE, ARTHUR H. M. und T. F. VERHOEF: *On the feasibility of situational method engineering*. *Information Systems*, 22(6-7):401 – 422, 1997.
- [vdML04] MASSEN, T. VON DER und H. LICHTER: *Deficiencies in Feature Models*. In: MANNISTO, TOMI und JAN BOSCH (Herausgeber): *Workshop on Software Variability Management for Product Derivation - Towards Tool Support*, 2004.
- [vG12] GERWE, D. VAN: *Wikipedia: Process-Data Diagram for Feature Driven Development*. Online, März 2012.
- [Wei12] WEIT E.V.: *Das V-Modell XT Online Portal*. Online <http://www.v-modell-xt.de/>, Mai 2012.
- [WH06] WILDE, T. und T. HESS: *Methodenspektrum der Wirtschaftsinformatik: Überblick und Portfoliobildung*. Arbeitsbericht des Instituts für Wirtschaftsinformatik und Neue Medien der LMU München, 2, 2006.
- [Woo12] WOOD, TOM: *Three old chestnuts, cracked*. Online, März 2012.

- [XR08] XU, PENG und BALASUBRAMANIAM RAMESH: *Using Process Tailoring to Manage Software Development Challenges*. IT Professional, 10(4):39–45, Juli 2008.
- [Zie08] ZIEGLER, ALEXANDER: *Eignungseinstufung von Vorgehensmodellen*. Doktorarbeit, Universität Bremen, 2008.