Interdisciplinary effects of technical debt in companies with mechatronic products — a qualitative study

Birgit Vogel-Heuser, Fandi Bi *

Institute of Automation and Information Systems, Technische Universität München, 85748 Garching b. München, Germany

ARTICLE INFO

ABSTRACT

Digitalization of products and production systems requires a fusion of mechatronic disciplines, where interfaces between mechanical, electrical, and software engineering are inevitable. The increasingly rapid pace of innovations in mechatronic systems triggers decisions being taken under time and cost pressure. At times, compromises in technical solutions are made, neglecting their long-term damage to the system. Technical debt (TD), a concept from software engineering, refers to short-term benefits that lead to long-term negative consequences, e.g., in the form of more difficult maintainability or evolvability. This also applies to mechatronic systems, yet the knowledge of TD characteristics and correlations in the interdisciplinary life cycle has only received little attention. This first comprehensive survey investigates TD in mechatronics systematically and across sectors. 50 experts, of whom 42% hold positions as department heads, from 21 renowned companies and 10 sectors in the German-speaking region supported this study with real scenarios where TD caused damage to their system. 94 informative TD incidents that were classified into twelve TD types were recorded, of which 2/3 have not yet been eliminated and posed a potential risk to the system. TD emerges most frequently in the first three stages of the life cycle, where the consequences rarely remain isolated at their source but are forwarded to later phases and disciplines in the life cycle. In contrast to the research focus in software engineering, the multi-domain analysis of mechatronic TD issues reveals that software engineers are most burdened by Requirements TD and Infrastructure TD in the interdisciplinary environment.

1. Introduction

In the age of production and service systems automation, the three disciplines of mechatronics perform in an interdisciplinary environment, where interfaces with others are inevitable. The term **mechatronics** refers to the synergetic interaction of the disciplines of mechanical, electrical, and software engineering in the design and manufacturing of industrial products as well as in process design (Bishop, 2002; Mori, 1969; Kyura and Oho, 1996). Previous studies of TD in mechatronics involved **automated Production Systems (aPS)**, which comprise mechatronic parts that are tightly interwoven (Bonfe and Fantuzzi, 2003; Rzevski, 2003). APS show characteristics that the software is strongly dependent on the hardware, with a long life cycle of up to 50 years (Vogel-Heuser et al., 2015b,a). For this study, we extend the range **from aPS** to general **mechatronic products** (cp. Fig. 1).

To increase the efficiency and decrease the consumption of resources, mechatronic disciplines aim for parallel development, whereas the associated complexity increases (Gausemeier and Moehringer, 2003). However, due to delays in mechanical and electrical engineering, this parallel development is not always feasible. Often, software engineering is the last activity in the development process, where it is faced with a lack of information and limited development time (Vogel-Heuser and Hess, 2016). Technological evolution introduces data and information processing into mechatronics, gradually transforming the traditional shop floor into an ecosystem, where networked systems interact with both the physical and organizational environment (Leitão et al., 2016). Despite the complexity within each of the mechatronic domains, models as well as designs of their joint dynamics are required.

The term "technical debt" originates from software engineering, and describes technical compromises that are expedient in the short term, but create a technical context that makes future changes more costly or impossible (Avgeriou et al., 2016c). In mechatronic systems, technical debt items (TD items) have been revealed in both mechanical and electrical engineering, in addition to software engineering (Avgeriou et al., 2016a,b). The common practice in the development of mechatronic systems is that software development then starts, when the development of mechanical and electrical development is already at a stage where any changes are expensive and time-consuming (Thramboulidis,

* Corresponding author.
 E-mail addresses: vogel-heuser@tum.de (B. Vogel-Heuser), fandi.bi@tum.de (F. Bi).
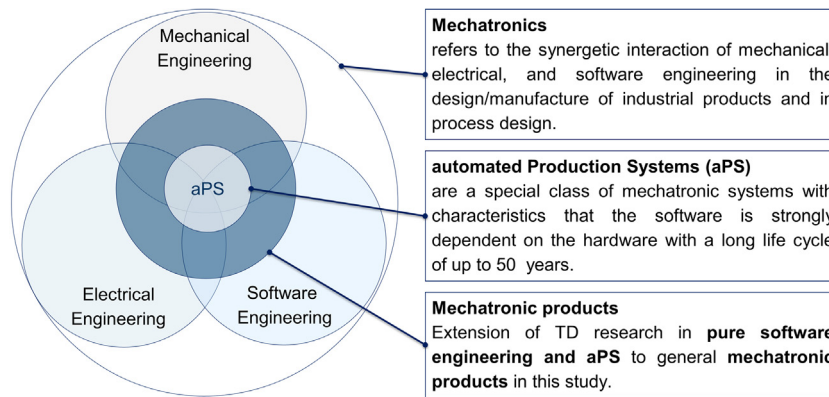
Mechatronics
refers to the synergetic interaction of mechanical, electrical, and software engineering in the design/manufacture of industrial products and in process design.

automated Production Systems (aPS)
are a special class of mechatronic systems with characteristics that the software is strongly dependent on the hardware with a long life cycle of up to 50 years.

Mechatronic products
Extension of TD research in **pure software engineering and aPS** to general **mechatronic products** in this study.

Fig. 1. Clarification of terms mechatronics, automated production systems (aPS) and mechatronic products.

2008). Therefore, mechanical and electronic properties impose many constrains and narrow the solution space for software development. Problems to be addressed in the mechanical or the electronic domain are solved by software, even architecture design problems in hardware parts. Malfunctions created by the mechanical and electrical disciplines need to be fixed by modifying the software on-site (Vogel-Heuser and Hess, 2016; Vogel-Heuser et al., 2015b). The industry maintains the assumption that manufacturing is expensive while the software is cheap, although this balance seems to be visibly shifting (Avgeriou et al., 2016a).

During the development of a mechatronic product, the requirements of software engineers are mostly subordinate to decisions made in mechanical and electrical engineering, e.g., about the movements to be executed by a mechanical part or which type of signals need to be processed. Software engineers might encounter challenges such as necessary information not being provided in a way that can be processed easily. In some instances, the information at hand is not specific enough or has inconsistencies. The lack of quantitative measures challenges the testing even more. This forces software engineers to work out the flowchart of a mechatronic product themselves. This kind of cross-disciplinary TD in multidisciplinary systems mostly appears due to the different disciplines operating primarily as "islands", optimizing only within the disciplinary boundaries and ignoring potential consequences for other disciplines (Avgeriou et al., 2016a).

Considering that TD in one discipline can affect others, questions arise of how the TD items correlate in the product development process and to each mechatronic discipline, as well as how the mechatronic disciplines are differently affected by TD. To provide a better understanding of the TD phenomenon in the mechatronic disciplines, this paper investigates *TDinMechatronics*.

TDinMechatronics comprises the research of TD in mechatronic systems, investigating TD items, causes, symptoms, consequences, patterns, and metrics for TD management. Not only is the focus on the extension of TD from research in software engineering to mechatronics. It is rather to understand the correlating interdisciplinary causes and effects.

In this study, the focus is set on understanding the occurrence and origin of *TDinMechatronics*, the correlating issues, and the reconstruction of TD characteristics. We conducted and analyzed 50 expert interviews in 21 companies producing mechatronic products, introducing a mechanic and electrical example as well as an interdisciplinary one for a better understanding of *TDinMechatronics*.

2. Background and related work

In this section, we first present the current state of research of TD in software engineering. To understand the additional challenges of *TDinMechatronics*, we map the life cycle of mechatronic systems (cp. Fig. 2) and explore current studies of TD in mechatronics.

2.1. TD in software engineering

The debt metaphor borrowed from the financial sector is firstly used in 1992 to communicate the inflexibility of an immature product caused due to limited development time (Cunningham, 1993). Since, the TD term received wide recognition and describes a collection of design or implementation constructs that are expedient in the short term but set up a technical context that can make future changes more costly or impossible (Li et al., 2015; Avgeriou et al., 2016c). Back then, the metaphor was referred to as coding practices. In contrast, nowadays, it includes broader facets across software development project lifecycles and communicates design trade-offs between developers and other decision-makers (Avgeriou et al., 2016c; Gat, 2010; Cunningham, 1993; Nord et al., 2012). As the system grows and evolves, these costs can become substantial and increase the burden on development teams. At the same time, if the debt is not repaid, the entire engineering organizations can be brought to a still-stand (Avgeriou et al., 2016c; Nord et al., 2012).

TD can be classified into different TD types according to its cause. The mapping of 94 studies of Li et al. resulted in ten coarse-grained TD types, where Code TD, Architectural TD, and Design TD have received the most attention (Li et al., 2015). While TD can be intentional or unintentional, yet the community shares different opinions on whether this characteristic is limited to specific TD types (McConnell, 2008; Nugroho et al., 2011; Martini and Bosch, 2015; Nord et al., 2012). The pervasive character of TD causes that it mostly occurs in combination and not in isolation (Avgeriou et al., 2016b; Rios et al., 2018a; Vogel-Heuser et al., 2015b).

2.2. TD in mechatronic systems

The term mechatronics made its first appearance in 1969 (Mori, 1969) and refers to "the synergetic interaction of the disciplines of mechanical engineering, electrical engineering, and software engineering in the design and manufacture of industrial products, as well as in process design.", which still appears to be valid (Kyura and Oho, 1996; Bishop, 2002; Gausemeier and Moehringer, 2003).

The TD research in mechatronics emerged in 2015 with the positioning study in regard to the applicability and adaptability
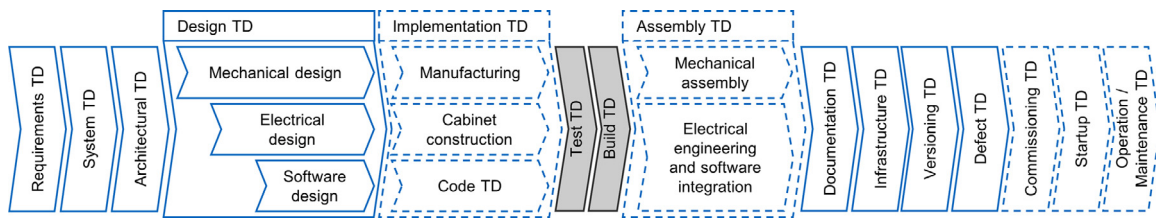
Fig. 2. Types of TD of software engineering matched and complemented to the life cycle of automated Production Systems (aPS); complemented elements by Vogel-Heuser and Rösch (2015) are indicated by dotted lines.

of TD for automated Production Systems (aPS) while identifying similarities and differences and adding specific challenges, causes, and effects (Vogel-Heuser and Rösch, 2015). Researchers in software engineering confirmed the existence of TD in mechatronics and the concept behind the TD metaphor by finding instances of various TD types that have been identified in the software engineering domain (Avgeriou et al., 2016a; Li et al., 2015). APS comprises mechatronic parts that are all closely interwoven and represent a special class of mechatronic systems (Rzevski, 2003; Bonfe and Fantuzzi, 2003). Therefore, the characteristics of mechatronic systems can be applied (Vogel-Heuser et al., 2015a). Comparing to pure software engineering, the software in aPS is strongly dependent on its hardware, whereas the interdependencies of the disciplines are to be seen as the most essential constraint for its development (Vogel-Heuser et al., 2015a). Therefore, cross-disciplinary interfaces are especially interesting and require increased focus (Vogel-Heuser et al., 2015b). However, these effects across disciplines are barely discussed by studies in the software engineering domain (Avgeriou et al., 2016a). Since, few studies have investigated *TDinMechatronics* based on heuristic approaches. The TD types of Li et al. (2015) were adapted to the life cycle of aPS and enlarged (Fig. 2) (Vogel-Heuser et al., 2015a; Vogel-Heuser and Rösch, 2015).

Few terms in the aPS' life cycle phases have domain-specific interpretation. Therefore, they are explained in the following:

- **Control Cabinet construction** *in electrical engineering includes both manufacturing of cabinets and terminal boxes housing* all electrical and electronic components (e.g., terminal blocks, power supply, frequency converters, often Programmable Logic Controller) connected according to the circuit diagrams designed.
- **Assembly** *in mechatronics:* the assembly of sub-parts of the own discipline to a part or system's integration with other disciplines.
- **Commissioning:**
  - *Software engineering:* installation of software on customer's devices.
  - *Mechanical, electrical engineering:* installation of product/production system at customer's premises.
- **Start Up**:
  - *Software engineering:* initial operation of the software on the customer's site.
  - *Mechanical, electrical engineering:* initial step-wise operation of the product/production system.
- **Maintenance:**
  - *Software engineering:* adaptive, perfective, corrective, and preventive measures on existing software to modify, adapt, correct, and improve software functionalities.
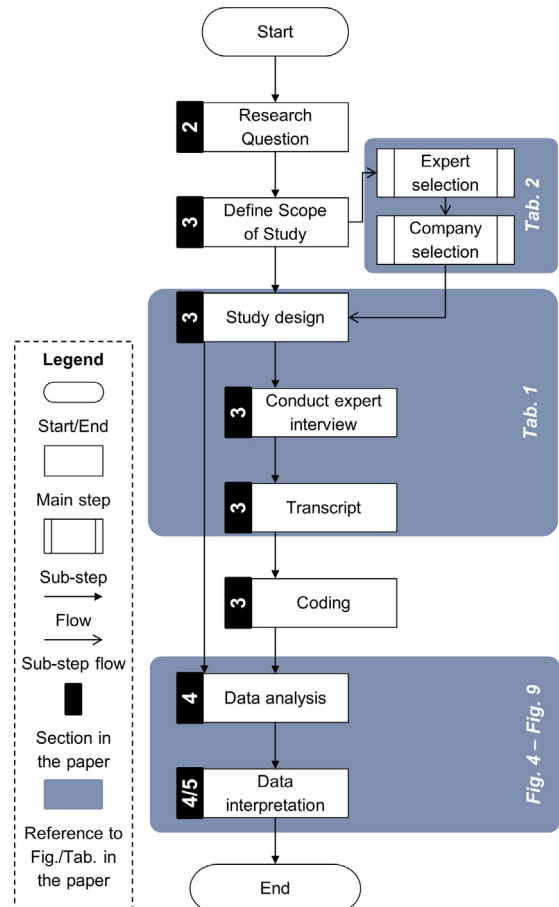

Fig. 3. Procedure of the study and correlating paper construct.

- *Mechanical, electrical engineering:* corrective and preventive measures including routine inspection, replacement of worn sub-parts, e.g., due to heat.

An investigation of TD was carried out in one company examining how TD is known, how much it "hurts," and how it is managed in aPS (Besker et al., 2017). The results show that "quite a lot of resources" in the form of extra-costs and its management is spent due to TD, while the company presents moderate awareness of what TD is and how much it is present in its systems. Most of the participants consider the average amount of wasted time as 32% of the development process, while Requirements and Test TD are ranked as the ones with significant impact (Besker et al., 2017). In a case study examining seven sites at five large software companies with embedded systems, different factors of Architectural TD have been identified, e.g., the pressure to deliver or prioritization of features over products (Martini and Bosch, 2017). In two typical case studies from a machine manufacturer

and a plant manufacturer, TD incidents were clustered, while TD types were extended (Dong and Vogel-Heuser, 2018). Time pressure, short-term cost-saving, and lack of experience were identified as the three most important reasons for sub-optimal solutions, while the software and mechanic discipline are those disciplines affected most by additional long-term effort. In 2019, an online survey based on a questionnaire analyzed TD in 48 German companies supplying aPS (Dong et al., 2019). It observes that the awareness for TD at these companies is generally low. While the electrical discipline is only partially affected by its own decisions, the software discipline takes the most benefit in taking TD. Comparable to the previous study, time pressure and short-term cost saving are the most chosen TD causes, followed by lack of experience and insufficient testing.

The management of TD in mechatronics requires extensive knowledge of the phenomenon, as the first step of TD Management is awareness, i.e., the identification of debt and its causes (Kruchten et al., 2012). *TDinMechatronics* was not yet examined across different industrial sectors, and it is unknown how different disciplines and their interfaces affect and are influenced by it. This first study focuses on mapping the current state of *TDinMechatronics* in the industry for a deeper understanding of its characteristics and pervasiveness. Specifically, we investigate the following research questions (RQ):

RQ 1: Which disciplines create the initial debt, and which disciplines are most affected?
RQ 2: Where along the life cycle are the TD created?
RQ 3: Are the individual mechatronic domains differently affected by the types of TD?

3. Research methodology

This section presents the case study design, clarifies the scope of the study, and explains how companies and experts were selected. Fig. 3 maps the case study procedure and correlates each procedure step with the referring section in the paper construct.

3.1. Scope of the study

To answer the RQs, information about confidential processes in the industry regarding TD is required. A sample size of 50 qualitative interviews is set to enable a picture of TD in the industry. This study identifies and collects qualitative *TD incidents* that increase the understanding of individual disciplines and their relation to TD. Each *TD incident* embodies a specific TD event. The recapture of *TD incidents* reflects the different aspects of the TD phenomenon in the experts' working environment. As the focus is on the occurrence of TD and correlating characteristics of *TDinMechatronics* (TD types, subtypes, causes, symptoms, and effects), this study aims to provide sound information for future management of interdisciplinary TD (TDM). However, activities of TDM are not explicitly examined in this study.

Although events in the future are dynamic, intentional decisions are initiated to bring specific future events with desired effects by present decisions. Compared to unintentional TD incidents, where the "present decision" is ambiguous, the comparison of desired effects to the occurred effects will not give grant the investigation of the cause–effect relations. Concerning the industrial TD incidents, this study focuses solely on intentional decisions that led to TD, as the causal chain of these decisions is more clearly traceable, and the time factor of the TD decisions made can be neglected (Avgeriou et al., 2016a).

3.2. Sampling strategy

Company selection — We focus on automotive and industrial automation as leading sectors in Ollero et al. (2006), but also examine a broad field of other sectors to determine whether TD is a cross-industry phenomenon. National and international companies had to cover the life cycle of at least one mechatronic product or production system to be included for this study.

Expert selection — Experts hold a special position in terms of type and quality of information available on a particular subject. To ensure the appropriate type and quality of information provided, an educational background in mechanical, electrical, or software engineering was required. Furthermore, the experts had to be directly involved in the life cycle of the product or production system produced. Hence the study excluded employees, e.g., in the purchasing department, sales department, and Chief Executive Officers (CEO) who were not directly involved in the development process. Experts had to confirm their work experience in the industry, which was challenging to measure. This study quantifies their work experience by using the duration of time spent working in mechatronics. At the time of the interview, they had to prove their work experience of at least two years. Experts can be classified as experts in a leading position, which refers to the position of a group manager or higher, or as specialists. Moreover, a classification is made according to the mechatronic discipline in which they work, as different backgrounds might affect the perception and different foci regarding TD. All experts in this study participated in their free will, and all statements regarding their activities and company are treated anonymously.

3.3. Expert interview

We selected the semistructured type of expert interview (Runeson and Höst, 2009), so the experts can qualitatively and quantitatively describe how they experience *TDinMechatronics*. The interview questionnaire features a combination of open and closed questions in four sections (cp. Table 1, first column). The number of questions was limited to 25 due to the limited interview time of between 30 and 45 min. The *first* section contained four open questions (#1–4) to gather the general information on the expert and the company, where the experts explained their previous knowledge of the term TD. The *second* section, with four open (#5, #9, #10, #13) and six closed questions (#6, #7, #8, #11, #12, #14), assessed information on the company's life cycle, e.g., characteristics of their interdisciplinary life cycle, variants/version management, and internal communication, which serve as the basis for evaluating company-specific correlations to TD traits. In the *third* section of the questionnaire, five open questions focused on TD-related topics (#15–19). In the *fourth* section, experts answered six closed questions about their opinion on *TDinMechatronics* (#20–25). Furthermore, Table 1 maps answer type, coding, and results to each question as well as shows which results were used to create the tables and figures of this study.

The interview and analysis procedure of this study was limited to six months, where the study progress was strongly dependent on external factors, e.g., the scheduling of the expert interviews or the approval process within the company. Therefore, a precise planning and systematic approach were applied to ensure the quality of the study outcomes.

3.3.1. Interview procedure

The expert interview is a direct method that belongs to the first degree of data collection techniques and offers most of the control for the researcher what form of data is collected (Runeson and Höst, 2009). Great importance was attached to the correct understanding of the question. Therefore, we decided to conduct all interviews in face-to-face sessions. After the expert has agreed on an interview, an introductory letter, a graphical representation of TD, and the interview questionnaire were sent for preparation purposes, and an appointment was arranged.

Prior to the start, permission was obtained from the expert to record the entire interview for a better recapture and reconstruction of the interview details. Furthermore, the semistructured interview procedure allows a more profound understanding of how the individual experts experience TD and allows that although the interview is planned, the questions are not necessarily asked in the same order as they are listed in the interview questionnaire. Depending on the conversation, the order of the questions is adapted to the conversation flow. All open and closed questions posed in the interview questionnaire were answered, where interviewees are allowed to expand all answers at any time (Runeson, 2012).

As proposed in the guidelines, only researchers that were involved during the interview have done the transcripts. The *member checking* method (Runeson, 2012), which allows material reviewed by the participants of the study, is applied. The interviewed experts get the opportunity to proofread the anonymized content, whether the level of abstraction and the understanding of the issue is accurate for them. It offers the experts the opportunity to clarify their answers, where suggestions are implemented.

3.3.2. Analysis procedure

With the fixed dataset, the data preparation phase follows. Considering the closed questions (cp. Table 1, column "Coding"), answers to these are limited to a selection of possibilities and can be coded as categorical or numerical data. Therefore, these answers to the closed questions are ready for further data analysis. Selected questions where "No coding" applied, serve the understanding of the organization structure, its operations as well as the working environment and tasks of the expert. These questions contribute to a better understanding and coding of the TD-related answers afterwards. Implicitly the answers of the "No-coding" questions are included in the coding of the TD-related answers. For the coding of the TD-related questions (cp. Table 1, Section 3), we applied the pattern coding method (Miles et al., 2014) (cp. Table 3) and performed further analysis to visualize the contagiousness (cp. Fig. 4). The coding categories on the top-level (TD types, TD sub-types, TD causes, TD symptoms, TD consequences, Initiating parties, and Affected disciplines) were derived from the questions #15–#19 (cp. Table 1). Existing categories and sub-categories of previous studies (cf. Li et al., 2015; Vogel-Heuser et al., 2015b; Rios et al., 2018b) were adopted, while the missing ones were enlarged. As previously introduced, the classification of Li et al. (2015) is coarse-grained. Thus, the classification of the TD incidents might be biased if only one researcher is involved in this classification. To address this threat, the *peer debriefing* method with other research colleagues is applied to lower the risk of the results being biased by one researcher (Runeson, 2012). In case of a mismatch of categorization, a third experienced researcher was consulted for the assessment of the specific case. Using two examples, we present our coding (cp. Table 3). This coding system was applied to all 94 TD incidents identified, which serves as the foundation of our data analysis.

4. Survey results

In this section, we summarize the results of our empirical study. Firstly, we show statistics describing the demographics of the participants, proceeding on to the examples and the quantitative data related to answer each RQ.

4.1. Descriptive statistics

Overall, we interviewed a total of 50 experts from the mechatronic disciplines of 21 companies (cp. Table 2). Out of these, we performed 96% of the interviews in face-to-face sessions, 2% in a video call, and 2% in an audio conference. Except for one expert, all others agreed to a voice recording of the full interview session. The leading position refers to the level of a group manager or higher, in which the overall rate is 64% in this study. Ten different sectors are represented in the study, with a focus on the sectors "automotive" (36%) and "industrial automation" (26%). As 78% of the experts in this study had no prior knowledge of TD, where 6% had it with a wrong interpretation, we could define the understanding of TD before and during the interview. The graduate process of exploring an issue with the expert in the semi-structured interview allowed a dialog, where we coined the experts' understanding of TD according to the Dagstuhl 16 162 definition (Avgeriou et al., 2016c).

4.2. Introduction of the terminology and coding by two TD incidents

To identify the most critical *TD incidents*, the 50 experts selected those that were the most representative to them. Overall, we gathered 94 *TD incidents* (cp. Table 2, sum of last column). Due to the pervasive character of TD, each TD incident could consist of multiple *TD items*. We identified 220 TD items and classified them into the respective TD type and TD sub-type. Besides, each TD incident could have one or more *TD causes, TD symptoms,* and *TD consequences*. This terminology aligns with the previously used terminology of Avgeriou et al. (2016c), adapting existing categories and sub-categories (cp. Section 3.3.2).

In this section, we introduce two examples of our case study for a better understanding of *TDinMechatronics*.

The *first* TD incident resulted from *Interview #29,* which involved mechanical and electrical engineering. We selected this case study due to its typical characteristics of challenges in interdisciplinary development processes. If there is a lack of processes to coordinate and/or audit the interfaces, additional expenses are incurred in the assembly phase.

Example I. "In the merging of mechanic and electronic parts in the second prototyping phase, an overlap was identified. Electronic cables block the fuse box and hinder it from opening. Although the entire mechatronic product is drawn in 3D, there is no 3D review process. Although there exists a "tracking of issues", no committee exists to decide which issue is to be fixed or not".

For the *second* TD incident, we selected one case that involved all three disciplines of mechatronics from *Interview #8*. Due to TD initiated in electrical engineering, all three mechatronic disciplines, as well as the customer, suffered from the significant negative consequences.

Example II. "To measure the fill level of fluids, fill level sensors are required. Some fill level sensors are reaching their limits with viscous fluids. Mainly due to lack of time, a sensor is chosen out of the experience. The specification of the customer was not studied carefully. Also, no research on the sensor's limits of performance was done. A sensor was selected based on the expert's experience. Due to the additional cleaning effort, a liquid test was

Table 1

Interview questionnaire – answer type, coding, result of each question and result presentation in the paper; *: contribute to a better understanding of TD-related results; ~: ambiguous questions – elaborated in discussion section (cp. 5.1).

#	Questions	Answer type	Coding	Results	Table 2	Table 3	Fig. 4	Fig. 5	Fig. 6	Fig. 7	Fig. 8	Fig. 9
I.	**General descriptive information about the person**											
1	What is your position and discipline in the company?	open answer	leading position - yes/no; position, discipline	Demographic data	x					x	x	
2	How long work experience in the industry /in a similar position do you bring with you?	years of work experience	2–9/9–19/20+	Demographic data	x							
3	Where do you see yourself in the product life cycle?	open answer	No coding	*								
4	Have you heard about "TD" before?	yes/no/open answer	yes/no/yes, but wrong interpretation	Demographic data	x							
II.	**General descriptive information about the company**											
5	Briefly explain your project process/development process.	open answer	No coding	*								
6	How would you cluster your projects (percentage distribution): new, adaptation, variant development?	%/%/%	percentage (new/adaptation/variant development)	*								
7	(Management) How many employees work under your leadership? (Specialist) How large is the team size?	number of person	number	*								
8	How do employees spread across the disciplines of mechanics, electrics and software in the team?	%/%/%	percentage (ME/EE/SE)	*								
9	How are the other disciplines involved in the collaborative development?	open answer	No coding	*								
10	How do you communicate within the team?	open answer	No coding	*								
11	How often does communication take place within the project team?	rating from 1–6	1–6	*								
12	How do you rate the quality of general information sharing in the team and why?	rating from 1–6	1–6	*								
13	Briefly explain your variant/version management – Do you use a tool?	open answer	No coding	*								
14	Are you in direct contact with other levels of the supply chain (e.g., OEM, Tier 1, Tier 2)?	other levels	OEM/Tier1/Tier2/End Customer	*								

(*continued on next page*)

not performed before the commissioning to the customer. Due to remaining defaults in the systems and on-site programming of partial machine coding, on-site problems were covered

After time-consuming troubleshooting, the sensor needs to be replaced. Mechanical engineering adapted the layout. Electrical engineering had to change the reading method, evaluation method, as well as the interpretation of its in- and output signals. The changed in- and output signals lead to necessary adaptations of the software code that processed the signals in a different way."

"An intensive research of the suitable sensor would have taken 40 h, yet now, the rework of all three disciplines require 100 h instead".

– Expert of Interview #8

4.3. Answer to RQ 1: Initial debt and the subsequently affected disciplines

Firstly, we start with the mapping of Example II, showing the contagiousness of TD throughout the mechatronic life cycle in a qualitative matter. Secondly, we analyze the disciplines initiating the debt with those that are subsequently affected.

4.3.1. Qualitative analysis of contagiousness of TD throughout the mechatronic life cycle

As one TD incident can contain different TD items, we investigated how these correlate to one another. In Example I, only one TD type (*Infrastructure TD*) was identified, which refers to the entire process and is therefore not mapped along the life cycle. However, by mapping Example II, we see that TD items spread out to later phases and "infiltrate" other disciplines (cp. Fig. 4).

Table 1 (continued).

| #	Questions	Answer type	Coding	Results	Table 2	Table 3	Fig. 4	Fig. 5	Fig. 6	Fig. 7	Fig. 8	Fig. 9
III.	**TD-related questions** (for each TD incident)											
15	Explain a TD incident in which you are/were involved in (TD incident, TD items in the TD incident, TD consequence).	open answer - TD incident	no coding	94 TD incidentsx			x	x	x	x	x	x
		open answer - TD item	TD type	220 items to TD types	x			x	x	x	x	
		open answer - TD item	TD subtype	220 items to TD subtypes	x							x
		open answer - TD consequence	TD consequence	15 TD consequences identified	x							
16	Which disciplines within the company were involved?	initiating discipline/party	ME/EE/SE/other parties	9 initiating parties	x	x						
		affected discipline/party	ME/EE/SE/other parties/none	4 subsequently affected parties	x	x						
17	Which disciplines/groups in the supply chain were affected by these decisions?	parties/groups in the supply chain	OEM/Tier1/Tier2/end customer/none	~								
18	What is the reason for the suboptimal decision to be made?	open answer - TD cause	TD cause	43 TD causes identified	x							x
19	How did one become aware of the suboptimal solution?	open answer - TD symptom	TD symptom	35 TD symptoms identified	x							x
IV.	**Final questions**											
20	Are there any control mechanisms in the company and have they been complied with?	open answer	No coding	~								
21	Do you see added value for your company to get the research results of the TD research in mechatronics?	yes/no/other	yes/no/other	47 yes/0 no/3 other								
22	Is there an awareness of TD in your organization, the conscious decision of a suboptimal solution?	yes/no/other	yes/no/other	33 yes/6 no/11 other								
23	In which discipline does TD cause the biggest long-term overhead?	open answer	No coding	*								
24	In which discipline could most of the effort be saved by the TD research in mechatronics?	open answer	No coding	*								
25	Could I contact you again for an online survey (15–20 min)?	yes/no/other	yes/no/other	49 yes/1 no/0 other								
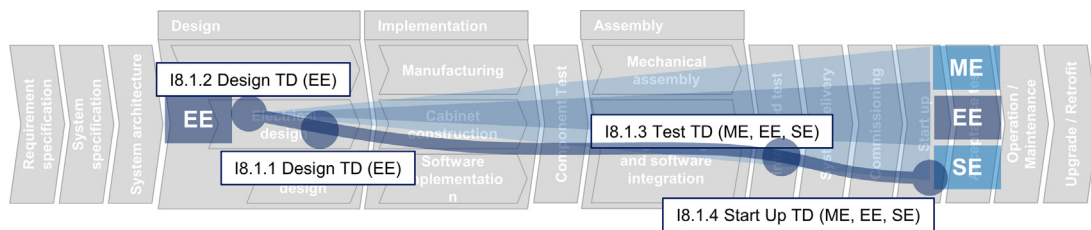
Fig. 4. Visualization of TD items of Example II in the correlating phases of the life cycle showing contagiousness of interdisciplinary TD; ME = Mechanical Engineering; EE = Electrical Engineering; SE = Software Engineering.

4.3.2. Quantitative analysis of initiating parties of the TD incident and subsequently affected ones

In the previous section, we saw that TD spreads out to later phases of the life cycle and involves more disciplines while it progresses. In this section, we analyze this aspect for all 94 TD incidents identified quantitatively. For this purpose, we use the coded data of the columns "Initiating parties" and "Affected disciplines" of Table 3. We identified nine initiating parties, and four subsequently affected ones. We started by sorting the initiating parties according to their sequence in the lifecycle (cp. Fig. 5, x-axis of the upper section). In 33% of the 94 TD incidents, we could observe that "Management" initiated the TD, while in 32%, the initial debt cannot are caused by multiple disciplines.

Table 2
Classified experts and companies included in the analysis.

Interview	Work experience in years	Position	Discipline[a]	Company size[b]	Industry[c]	Prior know-ledge of TD[d]	Number of TD incidents
1	20+	Head of R&D	ME	L	IA	N	2
2	20+	Head of Department	SE	M	MT	Y	2
3	20+	Head of Maintenance	ME	L	AV	N	1
4	20+	Head of Systems Engineering	EE	L	AM	N	3
5	2–9	Quality Engineer	EE	L	AV	N	4
6	10–19	Developer	EE	L	FI	Y	2
7	20+	Process Engineer	ME	L	AV	N	4
8	20+	Head of Department	ME	L	AV	N	3
9	20+	Head of Department	EE, SE	L	AV	N	1
10	2–9	Chief Executive Officer	ME	M	MT	Y	1
11	2–9	Maintenance Engineer	ME	M	MT	N	1
12	2–9	Chief Technology Officer	EE, SE	M	MT	N	1
13	10–19	Developer	SE	L	IA	YN	2
14	20+	Developer	SE	L	IA	YN	3
15	10–19	Developer	SE	L	IA	N	3
16	10–19	Group manager	EE	L	IA	N	1
17	2–9	Head of Department	ME, EE, SE	L	IA	YN	1
18	10–19	Head of Department	ME, EE, SE	L	IA	N	1
19	20+	Head of Department	ME	L	AV	N	1
20	2–9	Developer	ME	M	CV	N	2
21	20+	Chief Executive Officer	ME, EE, SE	M	MTM	N	4
22	20+	Head of Department	ME, EE, SE	L	FT	N	1
23	10–19	Head of Department	ME	L	FT	N	2
24	2–9	Head of Department	EE, SE	L	FT	N	5
25	20+	Developer	ME	L	FT	N	2
26	20+	Requirements Engineer	ME, EE, SE	L	IA	Y	1
27	2–9	Head of Department	SE	L	IA	Y	2
28	10–19	Head of Department	SE	L	IA	Y	1
29	2–9	Head of Department	EE	L	IA	N	2
30	10–19	Solution Architect	SE	L	IA	N	2
31	2–9	Head of Department	ME, EE, SE	L	ET	N	1
32	2–9	Project Manager	ME, EE, SE	L	ET	N	2
33	20+	Head of Department	SE	L	IA	N	3
34	20+	Head of Department	EE, SE	L	FT	Y	1
35	10–19	Head of Department	ME	M	FT	N	2
36	20+	Specialist in Manufacturing	ME	L	FT	N	1
37	20+	Head of Department	EE, SE	L	DE	N	1
38	20+	Head of Department	ME, EE, SE	L	AV	N	2
39	2–9	Quality Engineer	EE	L	CV	N	2
40	2–9	Quality Engineer	ME	L	AV	N	2
41	2–9	Project Manager	ME	L	AV	N	1
42	10–19	Project Manager	ME	L	AV	N	1
43	2–9	Project Manager	ME	L	AV	N	2
44	2–9	Project Manager	ME	L	AV	YN	2
45	2–9	Project Manager	ME	L	AV	N	1
46	10–19	Project Manager	ME	L	AV	N	1
47	20+	Project Manager	ME	L	AV	N	1
48	10–19	Group manager	ME	L	AV	N	1
49	2–9	Pre-Production Planner	ME	L	AV	N	5
50	2–9	Plant engineer	ME	L	AV	N	1

[a]Discipline: ME = Mechanical Engineering; EE = Electrical/Electronical Engineering; SE = Software Engineering.
[b]Company size: L = Large (over 250 employees); M = Medium (50–249 employees).
[c]Industry: AV = Automotive (36%); AM = Additive manufacturing (2%); FI = Food industry (2%); IA = Industrial automation (26%); MT = Measuring technology (8%); CV = Caravaning (4%); MTM = Machine & tool manufacturing (2%); FT = Forming technology (14%); ET = Environmental technology (4%); DE = Drive Engineering (2%).
[d]Y = yes (14%); N = no (78%); YN = yes, but wrong interpretation (8%).

Then, we ordered the four subsequently affected parties in the typical order of mechatronic product development (y-axis) (Thramboulidis, 2008). The combined analysis shows that the consequences of TD rarely remain isolated at their source; instead, they affect other parties/disciplines along the mechatronic life cycle (cp. Fig. 5, diagonal framed area). Furthermore, a related phenomenon is also observable that those parties/disciplines that are located later in the life cycle are more severely affected by TD than the ones located in the beginning (cp., Fig. 5, dotted area — dotted line draws the reflective axis of Management–Mechanical–Electrical–Software Engineering; more dots in the bottom part of the axis are colored in a darker shade than in the upper part). While issues initiated by mechanical engineering are decreasing in later stages of the life cycle, those produced by electrical engineering and software engineering seem to remain in their own system.

4.4. Answer to RQ 2: Occurrence of TD items throughout the life cycle and their causes

To investigate RQ 2, we use the coded data of the columns "TD type" of Table 3. The coding of the expert interviews shows that besides the 9/10 TD types of software engineering (Li et al., 2015), we could confirm *Start Up TD* and *Maintenance TD* from previous research of TD in aPS (Vogel-Heuser and Rösch, 2015). Since both TD types have no specified definition in the previous work, we apply the following definitions in this work.
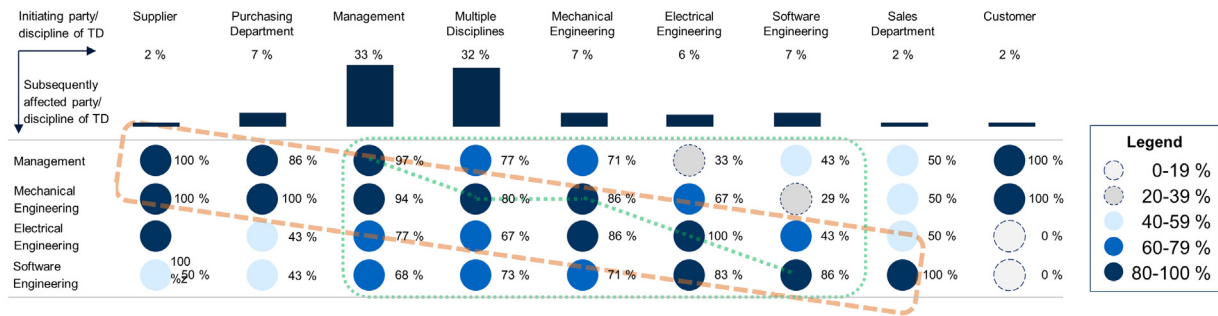
Fig. 5. Distribution of the number of disciplines with initial debt (x-axis) and consequently affected disciplines (y-axis) — sorted by the order of the supply chain and the mechatronic product development process; rounded to the nearest integer; n = 94.
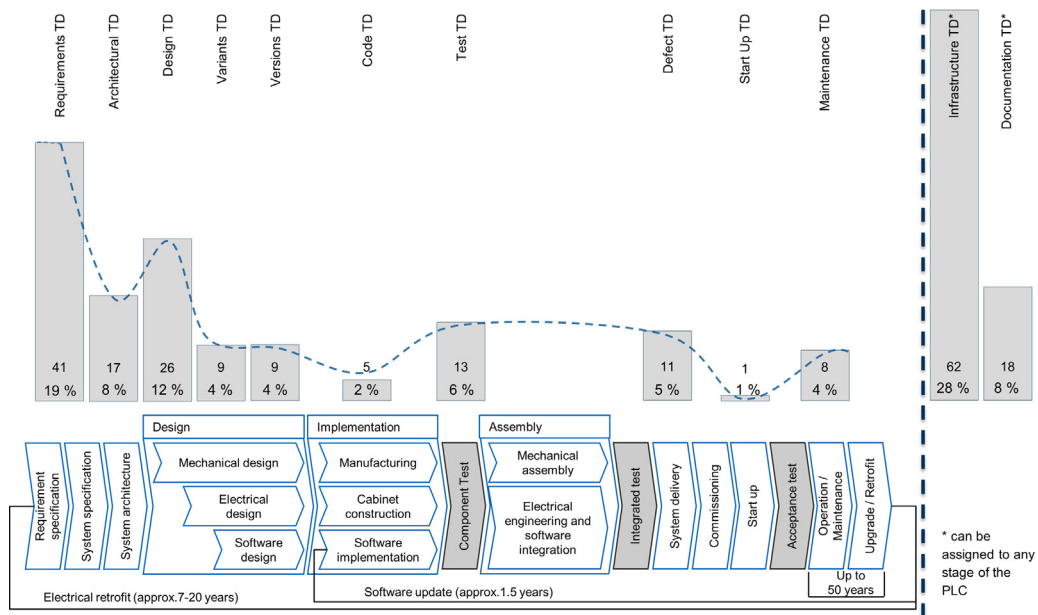

Fig. 6. Distribution of the number of classified TD types — sorted in the order of the mechatronic life cycle.

– *Start Up TD* refers to shortcuts taken in the startup process of the product or system. Such a shortcut negatively affects its quality, e.g., performance during operation.
– *Maintenance TD* refers to any kind of handicap with adverse effects on the product or system's maintenance.

Besides, TD aspects concerning the subject of variants, e.g., the building/reuse of variants or variant management, have not yet been covered. Although Versions TD might be clustered to Design TD, and Variants TD to Architectural TD, this does not pay respect to the importance of variants and versions in automation systems (Fischer et al., 2018). Therefore, we distinguish Variants TD and Versions TD in the scope of this study. Where Versions TD is established since Li et al. (2015), we introduce "Variants TD" to reduce the gap in research.

– *Variants TD* refers to flaws created while building, reusing, maintaining, or managing variants. Adverse effects are, e.g., unintended complexity, inconsistency, or redundancies.

Overall, the 94 TD incidents gathered involve a total of 220 TD items. We mapped them according to the sequence of the mechatronic life cycle (cp. Fig. 6). For the rearrangement, we referred to the order of the most common development process applied in mechatronics, the mechatronic V-Model, with extension to aPS (Vogel-Heuser et al., 2015b). In this representation, "Documentation TD" and "Infrastructure TD" are excluded from mapping, since the former can be related to every stage of the life cycle, and the latter can refer to the general process, tool, and method by its definition (Li et al., 2015). Fig. 6 shows that most of the TD types identified emerge in the early stage of the life cycle, and decline throughout the process. As some of the TD types are coarse-grained by their definition (Li et al., 2015) and comprise different topics (e.g., Infrastructure TD), Fig. 7 shows the number of TD items assigned to the more precise classification and understanding of TD sub-types correlating to the experts' discipline.

Besides, to offer a more in-depth understanding of the reason why the identified TD incidents were initiated, Fig. 8 shows the most frequently mentioned TD causes to the TD incidents (mentions in more than 8 TD incidents), with the interdisciplinary TD causes highlighted by "*". Notably, four out of the nine TD causes are of interdisciplinary nature, which was not yet considered in TD research of software engineering. Considering TD involving multiple disciplines, we shall not underestimate the interdisciplinary effects.
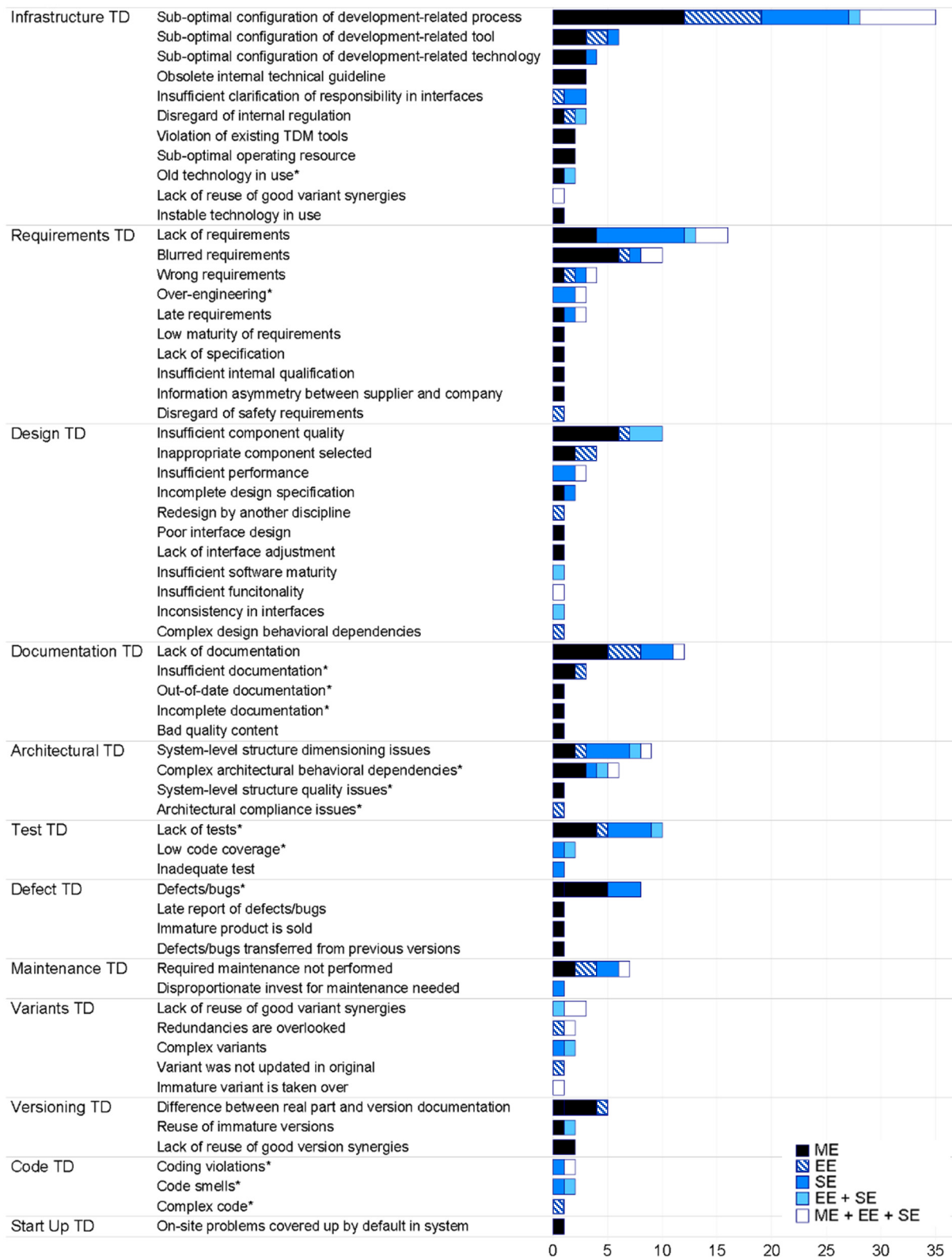
Fig. 7. Distribution of TD types and TD sub-types in correlation to discipline of the expert; $n = 220$ *TD items*. * TD sub-types introduced in Li et al. (2015); ME = Mechanical Engineering; EE = Electrical Engineering; SE = Software Engineering.
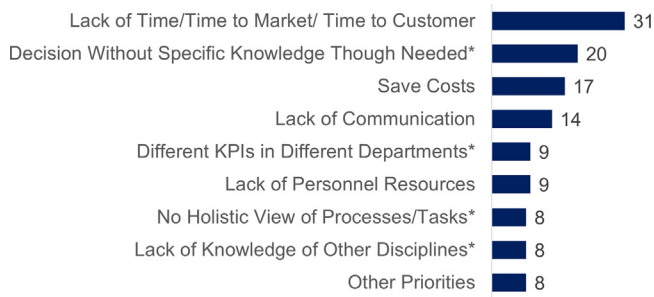
Lack of Time/Time to Market/ Time to Customer	31
Decision Without Specific Knowledge Though Needed*	20
Save Costs	17
Lack of Communication	14
Different KPIs in Different Departments*	9
Lack of Personnel Resources	9
No Holistic View of Processes/Tasks*	8
Lack of Knowledge of Other Disciplines*	8
Other Priorities	8

Fig. 8. Most frequently mentioned TD causes of the 94 identified TD incidents; * interdisciplinary TD causes.

4.5. Answer to RQ 3: Correlations of mechatronic disciplines to different TD types

To investigate RQ 3, we mapped the expert's background in correlation to the TD types along the life cycle. Fig. 9 shows the discipline of the expert correlating to the frequency of each TD type mentioned.

The experts interviewed work in different disciplines. Therefore, we clustered them into five categories — "Mechanical Engineering", "Electrical Engineering", "Software Engineering", "Electrical Engineering + Software Engineering", and "Mechanical Engineering + Electrical Engineering + Software Engineering" (cp. Fig. 9). In this study, each expert is only allowed to be assigned to one category, as multiple assignments would bias the results and mitigate the construct validity when one factor

affects another. Depending on the experts' perception, Fig. 9 shows which TD type receives the most attention. The "n" in the first column indicates how many experts were classified into each category, while sum of each row is 100%, representing the experts' perception of each category. Likewise, it is possible to see whether some TD types are perceived similarly or differently across the disciplines.

These results give insights into the different foci of each discipline. Although Infrastructure TD is perceived similarly across all discipline categories, other TD types vary considerably more. Requirements TD is most represented in software engineering and the consideration of all three mechatronic disciplines, while Design TD stands out in the interdisciplinary setting of electrical and software engineering. These results offer statements, which contradict previous research focused on TD in software engineering. Based on previous knowledge from the literature, we have assumed from the research distribution that Code TD, Architectural TD, and Test TD are those TD types that most burden experts in software engineering. Yet, the distribution of Fig. 9 shows clearly that in case software engineers are involved in an interdisciplinary life cycle of a mechatronic product. They are far more strained by Infrastructure TD as well as Requirements TD. Requirements TD, Architectural TD, and Design TD result in 60% of the gathered TD types, whereas the other seven share 40% in the later stages.

4.6. Frequent characteristics in expert interviews

During different expert interviews, we were able to record repetitive paradigms regarding TD. In the following paragraphs, we present quotes that describe the experts' challenges cooperating across disciplines.
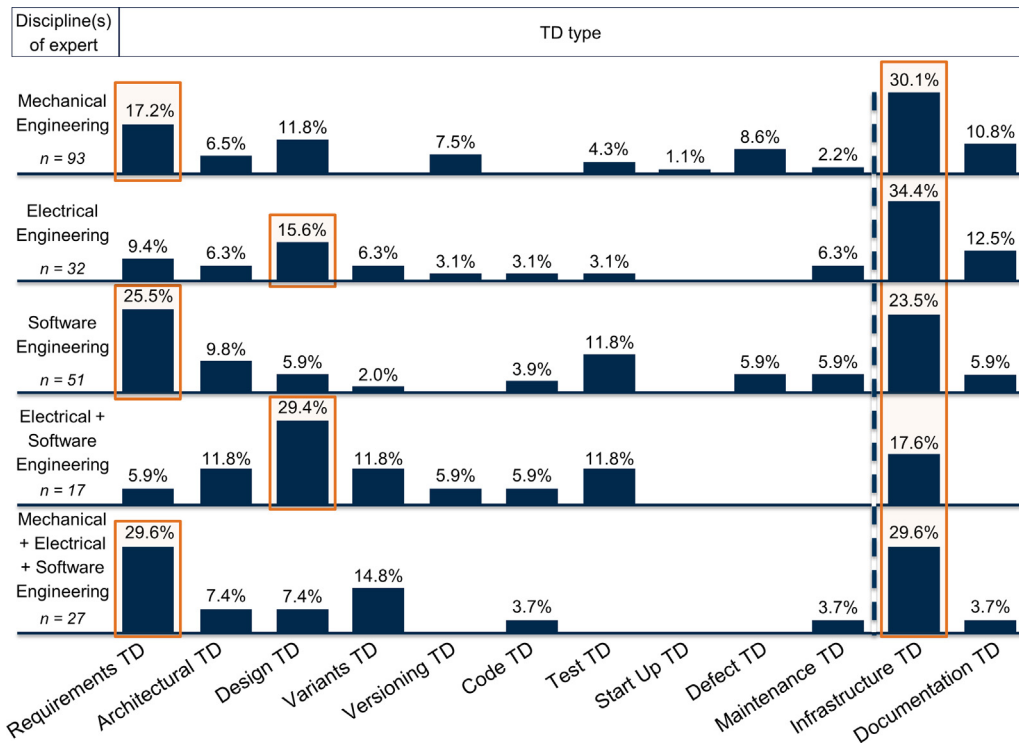

Fig. 9. Distribution of the number of classified TD types sorted in the order of the mechatronic product development process cp. Fig. 6; each TD item is only assigned once to the expert's discipline(s) and TD type.

Table 3
Coding sample of an interdisciplinary TD incident including seven TD items.

#Interview	#TD incident	#TD item	TD item - description	TD type	Reason for TD type selection	TD sub-type	TD cause[a]				TD symptom[a]			TD consequence[a]	Initiating parties	Affected disciplines[a]			
							Lack of time/time to market/time to customer	Save costs	Decision without specific knowledge though needed	Other priorities	Suboptimal performance	Error	Time-consuming trouble-shooting on-site	Rework		ME	EE	SE	MA
8	1	1	The sensor was selected without specific research on its limits of performance.	Design TD	Sensor is not suited for viscous liquids	Inappropriate component selected	1				1			1	EE	1	1		
8	1	2	Datasheet of the supplier was not studied carefully. A precise internal specification would have detected the inconsistency.	Design TD	Insufficient technical solution despite correct requirement	Incomplete design specification			1		1			1	EE	1	1		
8	1	3	A wet test should have been performed before handing it out to the customer.	Test TD	Extra effort in testing needed and not performed	Lack of tests	1	1	1				1	1	EE	1	1	1	
8	1	4	The initial run took place at the customer's site. Due to faulty sensor selection and lack of initial commissioning in the factory, assembly errors and wiring errors could not be noticed in time.	Start Up TD	Initial run takes place at the customer's site	On-site problems covered up by default in system	1	1					1	1	EE	1	1	1	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮				⋮			⋮	⋮				
29	1	1	Lack of 3D review process of the interdisciplinary product, 3D review only exists for components.	Infrastructure TD	Lack of process step to identify inconsistencies	Sub-optimal configuration of development-related process				1	1			1	Multiple disciplines	1	1		1
29	1	2	A list of issues already exists and has already contents of three pages. However, there is no committee with responsibility to decide	Infrastructure TD	Lack of a committee that is responsible for decisions	Sub-optimal configuration of development-related process				1	1			1	Multiple disciplines	1	1		1

[a]Multiple assignments possible; ME = Mechanical Engineering; EE = Electrical Engineering; SE = Software Engineering; MA = Management.

Agile development methods

Agile methods are applied to the development of mechatronic products. However, this interdisciplinary adoption also poses challenges and shares different interpretations across organizations:

- Different projects might be coordinated differently within one company. Some projects work with a traditional project manager, and others work with the SCRUM methodology. Also, while working with SCRUM, the sprint duration varies from team to team that can cause TD in interdisciplinary interfaces. (Interview #7)
- SCRUM is used for interdisciplinary development. Every sprint takes two weeks, and the sprints are in parallel for each discipline. Yet, there is a big problem due to the agile way of development. It remains unknown how the final mechatronic product will turn out. Therefore, often the test cases have to be adapted to the final product at the end of the process that can be cause to Test TD. (Interview #39)
- SCRUM is applied for the interdisciplinary development. There are daily standups at the plant. The core team is always present, and every team member can participate without any registration. The average number of participants at a daily standup is 15 members. This frequent exchange is established in the development process to mitigate TD. (Interview #49)

Collaboration between disciplines and departments

The experts addressed their concerns regarding the collaboration and communication within their organization.

- There is a big issue concerning change management between the disciplines of mechanical engineering and electrical engineering. It is more competitive than cooperative engineering. This may be the reason why some disciplines have to repay the TD that is caused in other disciplines. (Interview #6)
- Not all departments are involved in the process since the beginning; some related departments are not invited for discussions in the planning phase. Infrastructural TD can occur due to sub-optimal configuration of the development-oriented process. For a holistic solution, all affected departments need to be involved. (Interview #20)
- The quality and quantity of communication highly depend on the priority of the project. The communication is excellent when the project is the first priority. However, as soon as the priority is in second place, communication is insufficient and TD is caused. (Interview #18)
- The communication within the team seems to be good, but there are evident deficits regarding communication outside the department that induce TD, most of all along the product life cycle. (Interview #18)

Imbalanced level of information and power

Although imbalanced levels of information and power might exist due to positions in the supply chain, e.g., customer–supplier relationships, late changes in requirements should be avoided, for in-house and external developments. The caused consequences may generate ripple effects along the life cycle and implicate new TD items.

- It is an accepted practice for a vehicle manufacturer to make changes in requirements at a relatively late stage of the product development process. Intentionally, Requirements TD is caused. (Interview #35)

- In the lower levels of the company, the consciousness of TD is already present. Further upwards, one has the impression that there is no awareness of what is being done. Alternatively, perhaps TD decisions are made consciously, but then, it is not communicated to the development engineers properly. If the concrete reason for a decision is forwarded to the development department, the specialist would be better able to understand it. (Interview #13)

5. Discussion

This section interprets the procedure and results of the expert analysis. It elaborates on their implications for researchers and practitioners that work in a mechatronic environment.

5.1. Findings in TDinMechatronics

TD Types

In this study, we were able to identify nine of the ten TD types of Li et al. (2015) from software engineering, except for Build TD. Two types of TD from the aPS have been detected, Start Up TD and Maintenance TD (Vogel-Heuser and Rösch, 2015).

Software engineers may see the design, versioning, and variants as a part of the system's architecture. However, the term architecture in mechatronics deviates in its understanding, while fundamental methods supporting the variability of modular aPS are still limited to the software domain (Vogel-Heuser et al., 2015b). In aPS, new variants (V_B) are still often derived from existing ones (V_A). Due to the parallel operation of the different disciplines as well as with different customers at different sites, faults in variant V_B may occur before they arise in the initial variant V_A. They should be fixed in both variants, though, necessitating knowledge about variants and versions as well as configuration management (Vogel-Heuser and Ocker, 2018). Previous studies in aPS reveal that companies have substantial difficulties managing complexity, versions, and variants, especially when multiple disciplines are involved (Vogel-Heuser et al., 2017). A survey of 68 companies from machine and plant manufacturing (Vogel-Heuser and Ocker, 2018) shows that unfortunately, 44% of the companies do not use any tool to support variant management. Besides, 30% of the participating companies do not implement version management at all. Until today, modularity, reusability, variant, and version management is critical to the mechatronic development processes (Vogel-Heuser and Ocker, 2018). For this reason, we introduce "*Variants TD*" as a separate TD type in this study for an in-depth elaboration and future management.

Infrastructure TD was the TD type mentioned the most by experts. However, Infrastructure TD, according to its definition, is a "sub-optimal configuration of development-related processes, technologies, supporting tools, etc." (Li et al., 2015). This definition unites a broad range of notions, although the processes, technologies, and supporting tools might show different properties. Therefore, a more detailed classification than only on the TD sub-types level needs to be targeted. Besides, as Infrastructure TD and Documentation TD could not be assigned to distinct life cycle stages, a method of their quantification is required for later TDM.

We could observe that on the surface, the phenomena of TD in software engineering overlap with those in mechatronics; nevertheless, with more in-depth analysis, the characteristics deviate strongly from the previous knowledge of TD in the software domain.

TD in the mechatronic disciplines

Mechanical engineering is affected at a rate of 100% by TD created in the purchasing department. Different factors, e.g., the disregard of follow-up costs and the products' final quality, lead to technical decisions harming the system. On the one hand, mechanical engineering is burdened by debt created in non-technical disciplines. On the other hand, it is observable that the TD incidents initiated in mechanical engineering dissolve in later stages of the life cycle. These may be transferred to follow-up disciplines to solve, i.e., electrical and software engineering. The suggestion for mechanical engineers would be to support the purchasing department with their expertise through closer cooperation. Furthermore, mechanical engineers might identify the many interfaces with other disciplines, including changes, requirements, and functionalities, e.g., software engineers need unambiguous and measurable objectives for testing. Without a cooperative definition of the target values, the software engineering department would need to generate them according to their perception or gather information from other sources, e.g., the manual.

Electrical engineering suffers at 100% from TD that they initiate in their discipline while the highest number of TD type initiated is Test TD. Being the discipline connecting the hardware and the software, it is burdened by 86% of the TD initiated by mechanical engineering, and 43% by software engineering. However, it forwards 67% and 71% to them. In the life cycle, Design TD receives the most significant attention with 29%, which receives even greater attention when considered in combination with software engineering. The design and adaptation of electric/electronic components under real environmental conditions require a high level of expertise and careful research. Here, tests and audits could cushion pervasive TD incidents, bad performance, and defects in the process. However, Test TD has increased over three times when working interdisciplinary with software engineering.

Software engineering in an interdisciplinary environment receives 71% of the debt initiated by mechanical and 83% by electrical engineering. Unlike the research focus of previous studies that focus on Code, Architecture, and Test TD (Li et al., 2015; Ernst et al., 2015), software engineers mention Requirements TD and Infrastructure TD as the most salient TD types. More surprisingly it is, that they do not initiate Requirements TD, and Code TD only receives 4% of the attention. Those TD incidents initiated by software engineering affect mechanical engineering at 29% and electrical engineering at 43%, compared to the high quote of 86% remaining in the software engineering itself. Software engineering might also focus on interfaces to other disciplines up front, e.g., clarifying the interfaces concerning the prerequisites on the type of the required information, or introducing approaches protecting themselves from taking over TD created in other disciplines and preventing TD. In software engineering, 40% of the experts have not heard of the TD concept (Martini and Bosch, 2017). However, the quote concerning mechatronic experts is significantly higher in this study of 84%. 74% of the experts stated after the TD phenomenon had been defined that TD decisions are made despite being aware that these decisions would cause extra effort over time.

TD in the management

By analyzing the TD items, we identified that management decisions initiate 32% of them. This number resulted, although 64% of the interviewed experts held management positions. The analysis also shows that in 97% of these issues, adverse effects remain a management issue in the subsequent course of the process. Therefore, the more TD incidents the management avoids initiating, the less they will be burdened by them afterwards. For experts in management positions, we suggest to explicitly clarify interfaces, requirements, and tasks alongside specialists from different disciplines involved in the final system to prevent organizational silo effects (Tamburri et al., 2015).

TD in the phases of the life cycle

The mapping of the identified TD types to the life cycle of mechatronic systems shows that 60% of the named TD types emerge in the first three stages, while the remaining ones share the last seven. With the visualization of contagiousness (cp. Fig. 4, comparable to Martini and Bosch, 2017), we could detect that the contagious characteristics of TD are not limited to Architectural TD (Martini and Bosch, 2015); moreover, the pervasiveness seems to be generalizable to mechatronics as a whole. Not only are developers forced to introduce additional TD due to existing TD (Besker et al., 2018), the management is affected as well. The effects of one TD item can be the cause of another in *TDin-Mechatronics*. For instance, Infrastructure TD appears at 81% in combination with another TD item, and Documentation TD at 100%.

5.2. Results compared to previous studies

TD cause, TD symptom, and TD consequences

The investigation of TD causes, symptoms, and consequences (effect) is essential to TDM, which was also an item of this study. However, the acquired data from the expert interviews reveal the following problems: (1) *the reported TD causes/symptoms/ consequences are linked to each TD incident the expert chose to report.* Therefore, this number would represent neither the frequency nor the criticality of these characteristics and requires further analysis. It could be implemented in a study where the experts get an expanded list to rate the characteristics as a representation of their company values. (2) *Understanding the cause–effect relation is crucial, not causes and effects separately.* Proved by previous case studies (Rios et al., 2018b), causes and effects might be identical. This phenomenon was also identified in this study. The TD symptoms and/or TD consequences of one TD item could be the TD cause for initiating the next TD items within the TD consequence. Thus, depending on the "cut" in the timeline, you may encounter some of the characteristics as TD causes, symptoms, or consequences. This is the reason why we are skeptical of the separate analysis of causes and effects in a "*probabilistic cause–effect diagrams*" (Rios et al., 2018b), as it conceals crucial details of the interaction within the TD incident. Due to the challenges above, we decided to report the most mentioned TD causes solely to avoid false or biased conclusions.

The analysis of the most mentioned TD causes in this study shows that four out of nine are of interdisciplinary nature. These results point out the need for research of TD in the cross-disciple TD incidents and the interfaces between parties/disciplines.

Quantification of TD characteristics

Similar to prior studies on quantification of TD for TDM, we had encountered issues when we asked the experts to approximate the principal and interest of their TD incidents. The estimation of extra effort measured in time or monetary units was proven nearly impossible for a single employee to estimate. The reason here is that most of the interviewed experts share a subjective perspective on the TD incident *and* only participate in a limited section of the life cycle. Hence these values of estimated extra effort would disregard additional influencing factors in the organization and industry. One expert claimed that due to the limited perspective of the process, it is hard to tackle the root cause of TD (Interview #7). This approves the statement that costs and benefits are often difficult to be monetized in practice (Guo et al., 2016; Klinger et al., 2011).

6. Threats to validity

Yin's four aspects of validity (Yin, 2013) are discussed in detail in the context of the threats to validity identified concerning our case study, and which may jeopardize the validity of the results.

6.1. Construct validity

Construct validity refers to the extent to which selected measures in the study represent the research questions to be investigated. To mitigate this threat, detailed information about the study was given to the experts before the interview. Besides, specific definitions related to TD and the life cycle were clarified upfront and during the interview. To obtain the most reliable results possible, 96% of the interviews were conducted in face-to-face sessions, 2% via video conference, and 2% via audio conference. Great importance was attached to the correct understanding of the question. In case the question was misinterpreted, it was rephrased to clarify the concept carefully in an unbiased manner. In the case study design, two interview questions were ambiguous. Experts had problems answering them, as the answers depend on the situation and perspective. In Q17 (cp. Table 1 field marked as "~"), experts had problems defining themselves in the supply chain. The terms, e.g., OEM, Tier 1, or end customer, are standard terms in the automotive industry. However, when it comes to industries, e.g., industrial automation, these terms become blurred. In case TD characteristics are to be investigated along the supply chain, more universal terms need to be introduced as classification. In Q20, we explore whether control mechanisms exist in the company. Due to multiple TD items within a TD incident, the control mechanism is likely to be different ones serving different purposes. However, as the transcription, coding, and classification of TD items are done after the interview, the complete mapping of specific control mechanisms to the yet unknown TD items is rarely possible.

6.2. Internal validity

Internal validity refers to the accuracy of inferences, e.g., if one factor affects another, which is to be investigated. The multi-track recruiting of the experts aimed to keep the selection of experts as objective as possible. To include companies that develop, produce and use highly complex mechatronic products, this study deliberately focused on the automotive and industrial automation sectors, with 63% of the participants. Therefore, no industry-specific analysis was carried out. The spectrum of experts from 10 industries was used to provide conclusions about mechatronic systems. To address the uneven distribution of the experts' backgrounds, all analyses of correlations concerning the disciplines of the experts are given a percentage value. Moreover, the structure from Li et al. (2015) is very coarse-grained, according to the authors themselves. The TD types do not exclude each other explicitly, so this may lead to overlapping categories in the classification.

6.3. External validity

External validity refers to the capacity to generalize the findings. A sample of 50 interviewees from the industry allows only limited generalizability. However, the question of what number would represent a sufficient sample size is arguable. In this study, we have broad coverage of medium and large companies, as well as the distribution of interviews across disciplines and domains. Phenomena are corresponding to TD artifacts in mechatronic domains, and we were able to identify 9 of 10 TD types from previous literature on software development. In this study, the TD incidents and the TD types gathered do not directly correlate to the number of TD incidents of the particular experts' companies. The experts presented TD incidents that, in their view, were most relevant and significant. Phenomena from industry can have multiple aspects. The TD incidents reflect the experts' subjective and partial perspectives. Therefore, the answers to the interview questions may be biased, and more aspects could influence the TD type classification of the incident. Snowball effects might lead to effects experts could not foresee. To mitigate this threat, 64% of the experts hold a management position that gives them a better overview of projects and processes compared to specialized experts.

6.4. Reliability

Reliability is concerned with to what extent the specific researcher influences the data, the analysis, and the results might be generalized. Deliberately, no concrete example of possible TD that could be classified as a TD type was introduced to the expert before or in the interview. The reference to an example could have influenced the interviewees' interpretation of TD and hence guided the issue they presented in a particular direction. Since one researcher conducted all interviews, the *peer debriefing* method was applied. A group of researchers was involved in reviewing the study and the classification into the TD types. 98% of the interviewees agreed to record the entire interview, which improves the transparency and comprehensibility of discussions and results.

Additionally, *member checking* was applied, which allowed the interviewed experts to review the transcripts to agree that the captured information represented their intention. The transcripts were updated according to their wishes, which only included minor changes. Thus, the transcripts represent the original opinion of the expert.

7. Conclusion and outlook

The research on TD in software engineering has received increased attention in the past two decades. However, few studies have analyzed *TDinMechatronics*. Previously, it was unclear how the cross-disciplinary collaboration and interfaces, as well as the expansion of the software development process to the entire life cycle of a mechatronic system, might affect the characteristics of TD. However, this information is essential to managing *TDinMechatronics*. Therefore, this first survey analyzed TD across sectors and examined how its characteristics differ from TD in software engineering. We are well aware that this study does not represent all TD types in *TDinMechatronics*, nor the coverage of all disciplines. The present research contains 50 expert interviews from 21 companies within 10 sectors exploring TD in mechatronic systems in an industrial setting. The semi-structured expert interview method is applied to ensure detailed conversations with experts and to understand the TD incident from its foundation, to discover where TD is created, what the causes are, and gain insight into how different TD types occur in companies. We identified "Variants TD" as one new TD type occurring in *TDinMechatronics*.

TD emerges in the first three stages of the life cycle more frequently. Not only do the TD types differ in cross-disciplinary comparisons, but different characteristics can also be obtained depending on whether a discipline is studied in isolation or in combination with others. Our results show that in interdisciplinary systems, the initial debt is rarely caused by one of the mechatronic disciplines. In contrast, management decisions cause TD in almost a third of the TD incidents captured in this study. This study shows that each party creates and suffer different

TD types. Initial debts of mechanical engineering disappear in the course of the life cycle, likely because subsequent disciplines are obliged to carry out rework. Additionally, the software community focuses mainly on Code, Architecture and Test TD; however, we identified that while software engineers collaborate with other domains as required in mechatronic systems, Infrastructure TD and Requirements TD seem to bother them the most.

This study exposes the contagious character of TD and verifies that TD might be incurred in one discipline, but burdens and has to be repaid in another discipline. Not only is it present in Architectural TD, but in software engineering as a whole and throughout the life cycle, crossing interfaces between disciplines and different parties in the supply chain. The fact of its contagious character highlights the need to prevent TD, as well as the repayment of the existing TD incidents. Although few experts interviewed were familiar with the term TD, it is present in their companies. 78% of the interviewed experts had never heard of TD before. Still, after the interview, 94% were convinced that the research results would provide them with a more reliable basis for making technically better decisions and that more attentiveness to the topic would reduce TD from occurring.

The future research of *TDinMechatronics* and its management can build on synergies of prior research in software engineering, as there are overlapping characteristics, e.g., TD types and TD causes. Nevertheless, future research should examine the interdisciplinary effects. Besides, the lack of coverage of TD types in software engineering should be expanded for *TDinMechatronics*. We must consider whether a discipline operates independently or is dependent on interfaces to other disciplines, where additional communication is required between engineers as well as mechatronic parts. For *TDinMechatronics,* each discipline focuses on particular TD types, which are perceived as problematic, and therefore individual strategies must be developed. Each expert has a limited point of view of a particular issue. Accordingly, more interviews are intended to be carried out within one company to obtain multi-disciplinary perspectives on one TD incident, as well as more reliable mechanisms that may be developed to gain quantitative TD measurements.

Although TD in software engineering show synergies to *TDinMechatronics*, the foci due to the interdisciplinary collaboration show evident deviations. TD causes, symptoms, and consequences need to be elaborated in cause–effect relations, possibly as sub-features of TD items. The analysis of the connection of TD items and their sub-features within a TD incident provides a possible approach for future TD identification and prediction. TD research in mechatronics requires more attention and further research towards its automated and systematic management to engender a healthier system and increased international competitiveness.

CRediT authorship contribution statement

Birgit Vogel-Heuser: Conceptualization, Investigation, Methodology, Resources, Supervision, Validation, Writing - original draft, Writing - review & editing. **Fandi Bi:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Visualization, Writing - original draft, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Avgeriou, Paris, Ernst, Neil A., Nord, Robert L., Kruchten, Philippe, 2016a. Technical debtbroadening perspectives report on the seventh workshop on managing technical debt (MTD 2015). SIGSOFT Softw. Eng. Notes 41 (2), 38–41. http://dx.doi.org/10.1145/2894784.2894800.

Avgeriou, Paris, Kruchten, Philippe, Nord, Robert L., Ozkaya, Ipek, Seaman, Carolyn, 2016b. Reducing friction in software development. IEEE Softw. 33 (1), 66–73. http://dx.doi.org/10.1109/MS.2016.13.

Avgeriou, Paris, Kruchten, Philippe, Ozkaya, Ipek, Seaman, Carolyn, 2016c. Managing Technical Debt in Software Engineering (Dagstuhl Seminar 16162). http://dx.doi.org/10.4230/DagRep.6.4.110.

Besker, Terese, Martini, Antonio, Bosch, Jan, 2018. Technical debt cripples software developer productivity. In: Nord, Robert L., Buschmann, Frank, Kruchten, Philippe (Eds.), Proceedings of the 2018 International Conference on Technical Debt - TechDebt '18. ACM Press, New York, New York, USA, pp. 105–114.

Besker, Terese, Martini, Antonio, Bosch, Jan, Tichy, Matthias, 2017. An investigation of technical debt in automatic production systems. In: Tonelli, Roberto (Ed.), Proceedings of the XP2017 Scientific Workshops on - XP '17. ACM Press, New York, New York, USA, pp. 1–7.

Bishop, Robert H., 2002. The Mechatronics Handbook, Second Edition - 2 Volume Set. CRC press.

Bonfe, M., Fantuzzi, C., 2003. Design and verification of industrial logic controllers with UML and statecharts. In: Proceedings of 2003 IEEE Conference on Control Applications, 2003. CCA 2003. IEEE, pp. 1029–1034.

Cunningham, Ward, 1993. The wycash portfolio management system. SIGPLAN OOPS Mess. 4 (2), 29–30. http://dx.doi.org/10.1145/157710.157715.

Dong, Quang Huan, Ocker, Felix, Vogel-Heuser, Birgit, 2019. Technical debt as indicator for weaknesses in engineering of automated production systems. Prod. Eng. Res. Dev. 13 (3–4), 273–282. http://dx.doi.org/10.1007/s11740-019-00897-0.

Dong, Quang Huan, Vogel-Heuser, Birgit, 2018. Cross-disciplinary and cross-life-cycle-phase technical debt in automated production systems: two industrial case studies and a survey. IFAC-PapersOnLine 51 (11), 1192–1199. http://dx.doi.org/10.1016/j.ifacol.2018.08.428.

Ernst, Neil A., Bellomo, Stephany, Ozkaya, Ipek, Nord, Robert L., Gorton, Ian, 2015. Measure it? Manage it? Ignore it? software practitioners and technical debt. In: Di Nitto, Elisabetta, Harman, Mark, Heymans, Patrick (Eds.), Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering - ESEC/FSE 2015. ACM Press, New York, New York, USA, pp. 50–60.

Fischer, Juliane, Bougouffa, Safa, Schlie, Alexander, Schaefer, Ina, Vogel-Heuser, Birgit, 2018. A qualitative study of variability management of control software for industrial automation systems. In: 2018 IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE, pp. 615–624.

Gat, I., 2010. How to Settle Your Technical Debt: A Manager's Guide. Cutter Consortium, Arlington, MA.

Gausemeier, J., Moehringer, S., 2003. New guideline VDI 2206 - A flexible procedure model for the design of mechatronic systems. In: Folkeson, A., Gralen, K., Norell, M., Sellgren, U. (Eds.), DS 31: Proceedings of ICED 03. pp. 599–600.

Guo, Yuepu, Seaman, Carolyn, Silva, Fabio Q.B. da, 2016. Costs and obstacles encountered in technical debt management – A case study. J. Syst. Softw. 120, 156–169. http://dx.doi.org/10.1016/j.jss.2016.07.008.

Klinger, Tim, Tarr, Peri, Wagstrom, Patrick, Williams, Clay, 2011. An enterprise perspective on technical debt. In: Proceeding of the 2nd Working on Managing Technical Debt - MTD '11. ACM Press, New York, New York, USA, p. 35.

Kruchten, Philippe, Nord, Robert L., Ozkaya, Ipek, 2012. Technical debt: From metaphor to theory and practice. IEEE Softw. 29 (6), 18–21. http://dx.doi.org/10.1109/MS.2012.167.

Kyura, N., Oho, H., 1996. Mechatronics-an industrial perspective. IEEE/ASME Trans. Mechatron. 1 (1), 10–15. http://dx.doi.org/10.1109/3516.491405.

Leitão, Paulo, Colombo, Armando Walter, Karnouskos, Stamatis, 2016. Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges. Comput. Ind. 81, 11–25. http://dx.doi.org/10.1016/j.compind.2015.08.004.

Li, Zengyang, Avgeriou, Paris, Liang, Peng, 2015. A systematic mapping study on technical debt and its management. J. Syst. Softw. 101, 193–220. http://dx.doi.org/10.1016/j.jss.2014.12.027.

Martini, Antonio, Bosch, Jan, 2015. The danger of architectural technical debt: Contagious debt and vicious circles. In: 2015 12th Working IEEE/IFIP Conference on Software Architecture. IEEE, pp. 1–10.

Martini, Antonio, Bosch, Jan, 2017. On the interest of architectural technical debt: Uncovering the contagious debt phenomenon. J. Softw. Evol. Proc. 29 (10), e1877. http://dx.doi.org/10.1002/smr.1877.

McConnell, Steve, 2008. Managing Technical Debt. Construx Software Builders, Inc, pp. 1–14.

Miles, Matthew B., Huberman, A. Michael, Saldaña, Johnny, 2014. Qualitative Data Analysis. A Methods Sourcebook. Edition 3. Sage, Los Angeles, London, New Delhi, Singapore, Washington DC.

Mori, T., 1969. Mechatronics. Yasakawa Internal Trademark Appl. Memo 21 (01).

Nord, Robert L., Ozkaya, Ipek, Kruchten, Philippe, Gonzalez-Rojas, Marco, 2012. In search of a metric for managing architectural technical debt. In: 2012 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture. IEEE, pp. 91–100.

Nugroho, Ariadi, Visser, Joost, Kuipers, Tobias, 2011. An empirical model of technical debt and interest. In: Proceeding of the 2nd Working on Managing Technical Debt - MTD '11. ACM Press, New York, New York, USA, pp. 1–8.

Ollero, A., Boverie, S., Goodall, R., Sasiadek, J., Erbe, H., Zuehlke, D., 2006. Mechatronics, robotics and components for automation and control. Annu. Rev. Control 30 (1), 41–54. http://dx.doi.org/10.1016/j.arcontrol.2006.02.002.

Rios, Nicolli, Oliveira Spinola, Rodrigo, Mendonca Neto, Manoel G. de, Seaman, Carolyn, 2018a. A study of factors that lead development teams to incur technical debt in software projects. In: 2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). IEEE, pp. 429–436.

Rios, Nicolli, Spínola, Rodrigo Oliveira, Mendonça, Manoel, Seaman, Carolyn, 2018b. The most common causes and effects of technical debt. In: Oivo, Markku, Méndez, Daniel, Mockus, Audris (Eds.), Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement - ESEM '18. ACM Press, New York, New York, USA, pp. 1–10.

Runeson, Per (Ed.), 2012. Case Study Research in Software Engineering. Guidelines and Examples, first ed. ebrary, Inc., Wiley, Hoboken, N.J.

Runeson, Per, Höst, Martin, 2009. Guidelines for conducting and reporting case study research in software engineering. Empir. Softw. Eng. 14 (2), 131–164. http://dx.doi.org/10.1007/s10664-008-9102-8.

Rzevski, George, 2003. On conceptual design of intelligent mechatronic systems. Mechatronics 13 (10), 1029–1044. http://dx.doi.org/10.1016/S0957-4158(03)00041-2.

Tamburri, Damian A., Kruchten, Philippe, Lago, Patricia, van Vliet, Hans, 2015. Social debt in software engineering: insights from industry. J. Internet Serv. Appl. 6 (1), 29. http://dx.doi.org/10.1186/s13174-015-0024-6.

Thramboulidis, Kleanthis, 2008. Challenges in the development of mechatronic systems: The mechatronic component. In: 2008 IEEE International Conference on Emerging Technologies and Factory Automation. IEEE, pp. 624–631.

Vogel-Heuser, B., Fay, A., Schaefer, I., Tichy, M., 2015a. Evolution of software in automated production systems: Challenges and research directions. J. Syst. Softw. 110, 54–84.

Vogel-Heuser, Birgit, Fischer, Juliane, Feldmann, Stefan, Ulewicz, Sebastian, Rösch, Susanne, 2017. Modularity and architecture of PLC-based software for automated production systems: An analysis in industrial companies. J. Syst. Softw. 131, 35–62. http://dx.doi.org/10.1016/j.jss.2017.05.051.

Vogel-Heuser, Birgit, Hess, Dieter, 2016. Guest editorial industry 4.0–prerequisites and visions. IEEE Trans. Automat. Sci. Eng. 13 (2), 411–413. http://dx.doi.org/10.1109/TASE.2016.2523639.

Vogel-Heuser, Birgit, Ocker, Felix, 2018. Maintainability and evolvability of control software in machine and plant manufacturing — An industrial survey. Control Eng. Pract. 80, 157–173. http://dx.doi.org/10.1016/j.conengprac.2018.08.007.

Vogel-Heuser, Birgit, Rösch, Susanne, 2015. Applicability of technical debt as a concept to understand obstacles for evolution of automated production systems. In: 2015 IEEE International Conference on Systems, Man, and Cybernetics. IEEE, pp. 127–132.

Vogel-Heuser, Birgit, Rösch, Susanne, Martini, Antonio, Tichy, Matthias, 2015b. Technical debt in automated production systems. In: 2015 IEEE 7th International Workshop on Managing Technical Debt (MTD). IEEE, pp. 49–52.

Yin, Robert K., 2013. Validity and generalization in future case study evaluations. Evaluation 19 (3), 321–332. http://dx.doi.org/10.1177/1356389013497081.