The Journal of Systems & Software 204 (2023) 111725

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Characteristics, causes, and consequences of technical debt in the automation domain[☆]

Fandi Bi [*], Birgit Vogel-Heuser [1], Ziyi Huang, Felix Ocker

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

ARTICLE INFO

Article history:
Received 16 November 2022
Received in revised form 26 March 2023
Accepted 20 April 2023
Available online 27 May 2023

Keywords:
Technical debt
Mechatronics
Mechatronic product
Life cycle
Causes
Consequences

ABSTRACT

Technical Debt (TD) is a significant concern in software development, particularly when interdisciplinary teams collaborate and interact. The goal of the study is to investigate TD causal chains and patterns in the industrial automation sector by analyzing 123 mechatronic TD incidents from 47 expert interviews across ten companies. Findings reveal that Requirements, Process, and Test TD are most common, while Build, Versioning, Manufacturing, Code, and Maintenance/Service TD are less frequent. Key causes include "other priorities", "lack of time", "historically grown products", "lack of market analysis" and "copy-paste-modify without revising tolerances." The research identifies correlations between TD subtypes and causes/consequences in relation to company size, experts' experience, and position, utilizing the Chi-square test and PrefixSpan algorithm. The study also maps the contagious character of TD using Neo4J graphical representation. This first in-depth analysis of TD causal chains in industrial automation contributes qualitatively to understanding TD patterns, helping researchers and practitioners assess TD contagiousness, comprehend its effects, prevent diffusion, and develop repayment strategies To the best of our knowledge, this study's quantitative analysis approach provides the foundation that will enable future research identifying TD metrics and TD management in multidisciplinary engineering.

1. Introduction

The increasing amount of data in today's technological domains has led to a greater emphasis on reasoning and learning from causal relations. This helps us better understand the key factors that influence a system, ultimately improving decision-making. One important concept in this context is "technical debt" (TD), which refers to 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 this study, we use the conceptual model and TD terminology developed by Avgeriou et al. (2016c) (cp. Fig. 1).

In industry, TD not only affects software engineering but also extends to mechanical and electrical engineering (Vogel-Heuser et al., 2015b). Decisions resulting in TD often stem from underestimated scope, impact, and remedial measures. Adaptive and flexible mechatronic systems require modular and reusable components, particularly due to the systems' long-term life cycle of up to 50 years (Vogel-Heuser et al., 2015a). However, the diversity of variants and modular concepts can lead to the spread of minor errors through copy, paste, and modify strategies, revealing TD's contagious nature in mechatronic systems (Vogel-Heuser and Bi, 2021).

Software engineers experience various TD types, depending on whether they work in isolation or interdisciplinarily. Isolated work focuses on investigating Code TD and Architectural TD; while interdisciplinary work deals more with Requirements TD and Infrastructural TD (Vogel-Heuser and Bi, 2021). Lazy compromises in early stages (e.g., unclear requirements due to not yet fully defined business cases, lack of documentation due to deadline pressure) can affect later stages, resulting in late changes and disproportionate investment in maintenance and troubleshooting. TD types differ in their frequency and impact on mechatronic companies (Bi et al., 2021), necessitating resource reallocation from ongoing projects for TD repayment.

TD's contagious character raises questions about its correlation during the product development process and its impact on different mechatronic discipline (Vogel-Heuser and Bi, 2021; Martini and Bosch, 2015). For instance, *Blurry requirements* (Requirements TD) lead to *Inadequate tests* (Test TD), allowing *Defects/bugs* (Defect TD) to remain undetected and pose a risk to the system. TD in one discipline can affect and harm other disciplines as well as later engineering stages. Thus, questions arise about how the TD

☆ Editor:Neil Ernst.

* Corresponding author.
E-mail addresses: fandi.bi@tum.de (F. Bi), vogel-heuser@tum.de (B. Vogel-Heuser), ziyi.huang@tum.de (Z. Huang), felix.ocker@tum.de (F. Ocker).

1 Core Member of MDSI and Member of MIRMI

https://doi.org/10.1016/j.jss.2023.111725
0164-1212/

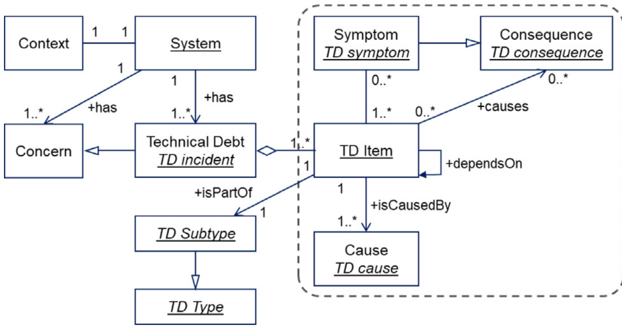

Fig. 1. TD conceptual model and terminology adapted from Avgeriou et al. (2016c); Underlined terms: terminology used in this study; Dotted area: relations later modified for coding and analyses. In this study, consequence/effect/impact (Avgeriou et al., 2016c) is summarized in the term consequence.

items correlate during the product development process and to each mechatronic discipline.

In addition, it is yet unknown what kind of causes lead to what kind of consequences. To better understand TD in the mechatronic disciplines, this paper presents a TD scheme and investigates the correlation of causes and consequences in 123 mechatronic TD incidents in the automation industry, a key sector undergoing digitalization. We extend the *TDinMechatronics* study (Vogel-Heuser and Bi, 2021) to 47 expert interview data in 10 companies, presenting 368 TD items, and identifying cause–effect relations. Through this research, we aim to uncover TD characteristics, TD patterns and metrics in mechatronic systems.

Specifically, we investigate the following research questions (RQ):

RQ 1. What are the most frequent TD types, TD subtypes, and TD causes/consequences experienced by experts in industrial automation resulting from 368 TD items?

RQ 2. What are the interdependencies behind the TD incident and their initiating and affected parties?

RQ 3. How do company size *(i)*, the experience of experts *(ii)*, and their leading position *(iii)* relate to the reported TD characteristics?

RQ 4. What patterns exist between TD types, TD subtypes, and TD causes/consequences?

The remainder of the paper is structured as follows: First, in Section 2, we present state of the art related to TD in software engineering and mechatronics, causes and consequences, and the effects of TD in mechatronic systems. Section 3 describes the case study design and methodology, detailing how we collected and analyzed data to answer the RQs. By examining confidential cases and processes in the industry regarding TD, we aim to uncover insights into the interdependencies behind TD incidents, their initiating and affected parties, correlations between different factors, and patterns among TD characteristics. Section 4 presents the results of our investigation, addressing RQ1 by discussing the most frequent TD types, sub-types, and causes/consequences experienced by experts in industrial automation. In addition, we explore the interdependencies behind TD incidents (RQ2), the relationship between company size, expert experience, and leading position with reported TD characteristics (RQ3), and patterns between TD types, sub-types, and causes/consequences (RQ4). In Section 5, we discuss the results, showing how causes and consequences affect each other and compare the results with previous studies' findings. Moreover, we discuss in detail the four threats to validity. The paper closes with Section 6 providing a conclusion, while an outlook regarding future TD correlations and metrics is given.

2. Background and related work

In this section, firstly, we compare the similarities of and differences between TD in software engineering compared to mechatronics. Furthermore, we focus on causal relations while elaborating on TD causes and consequences research.

2.1. Similarities of and differences between TD in software engineering and mechatronics

Since the metaphor "technical debt" emerged in 1992 to describe the inflexibility of an immature product due to limited development time (Cunningham, 1993), the term has been used to describe 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 (Avgeriou et al., 2016c). These costs can become substantial and burdensome within the growing and evolving system for development teams. Meanwhile, if the debt is not repaid, the entire engineering organization can be brought to a standstill (Avgeriou et al., 2016a; Nord et al., 2012).

Besides its applicability to the software engineering domain, the TD metaphor can be applied and adapted to systems with mechatronic characteristics, to the design and manufacture of industrial products, and to process design (Mori, 1969; Vogel-Heuser and Rösch, 2015). The use of the metaphor in mechatronics and its concept has been confirmed by researchers in software engineering, stressing that various TD types can be applied, and TD may further hurt other disciplines of the same iteration, i.e. design phase (Avgeriou et al., 2016a). The interactions of different disciplines, as well as the shorter and asynchronous innovation cycles of integrated products and production systems, result in unforeseen TD that is difficult to control. TD's main similarities and differences in software engineering and mechatronics are summarized in Table 1.

In software engineering, coarse-grained TD types are assigned to the TD item according to the cause, wherein Code TD, Architectural TD, and Design TD have received the most attention (Li et al., 2015). Most recently, Process TD and Security TD have emerged as the focus of research (Martini et al., 2020; Rindell et al., 2019). In mechatronics, four TD types need to be added due to the enlarged product life cycle phases, and the further disciplines investigated (Vogel-Heuser et al., 2015b; Vogel-Heuser and Bi, 2021; Dong and Vogel-Heuser, 2021b) :

- *Variants TD* refers to flaws created while building, reusing, maintaining, or managing variants.
- *Industrial Engineering TD* refers to shortcuts and flaws which occur during the industrial engineering phase, e.g., during production planning
- *Start Up TD* refers to shortcuts taken in the startup process of the product or system.
- *Maintenance/Service TD* refers to any handicap with adverse effects on the product or system's maintenance.

In addition to TD types' applicability, TD items in mechatronics, like those in software engineering, can have characteristics, e.g., causes, symptoms, and consequences (Avgeriou et al., 2016c). Yet, the pervasive character of TD types and causes mainly occurs in combination and not in isolation (Vogel-Heuser et al., 2015b; Martini and Bosch, 2015; Avgeriou et al., 2016b; Rios et al., 2018a). TD might be incurred in one discipline, but the burdens have to be repaid in another one (Vogel-Heuser and Bi, 2021).

Moreover, TD in mechatronics more often emerges in the first three stages of the life cycle, namely the requirement, architecture, and design phase (Vogel-Heuser and Bi, 2021) . Not

Table 1
Similarities of and differences between TD in software engineering and mechatronics.

	Similarities of TD in software engineering and mechatronics	Differences between TD	
		In software engineering	In mechatronics
General information	TD metaphor can be applied (Nord et al., 2012)	/	/'
TD type	TD type metaphor can be applied (Vogel-Heuser et al., 2015a; Nord et al., 2012; Li et al., 2015)	Built TD was not identified in mechatronics yet (Bi et al., 2021; Martini and Bosch, 2015)	Additional TD types were identified: Variants TD, Industrial Engineering TD, Manufacturing TD, Start Up TD, Maintenance TD (Vogel-Heuser et al., 2015a; Bi et al., 2021; Avgeriou et al., 2016b)
Most frequently appearing TD types	Design TD	Code TD, Architectural TD (Martini et al., 2020)	Infrastructure TD/Process TD, Requirements TD (Bi et al., 2021)
Contagiousness	Character of TD types and causes mostly occur in combination and not in isolation (Avgeriou et al., 2016c; Bi et al., 2021; Martini and Bosch, 2015)	Architecture TD is contagious (Cunningham, 1993)	All TD types are contagious – e.g. due to copy, paste and modify (Bi et al., 2021). TD occurred in one discipline might be repaid in another one (Nord et al., 2012)
Occurrence in phase of life cycle	TD Types can be assigned to different phases of the life cycle (Martini et al., 2020)	/	Most frequently in first three stages of the life cycle, requirement, architecture, and design phase (Bi et al., 2021; Martini and Bosch, 2015)
TD cause and TD consequence	TD cause metaphor can be applied (Li et al., 2015; Rios et al., 2020)	TD causes and consequences were investigated individually and not in causal relations (Mordal-Manet et al., 2009; Rios et al., 2020)	TD is rarely caused by one discipline (Bi et al., 2021). Most occurring TD causes and interdisciplinary TD causes have been identified (Bi et al., 2021; Martini and Bosch, 2015). Further details have not been investigated yet
TD Management	TD management activities can be applied (Li et al., 2015; Martini et al., 2020)	Tools exist to manage TD, mainly to address Code TD, Architectural TD (Lenarduzzi et al., 2019; Guzman et al., 2017; Holvitie and Leppanen, 2013; Wagner et al., 2012; Kontsevoi et al., 2019; Dong and Vogel-Heuser, 2021a)	No TD management tool exists yet, but only few concepts (Waltersdorfer et al., 2020; Biffl et al., 2019b,a; Tom et al., 2013; Dong and Vogel-Heuser, 2018)

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 (Bi et al., 2021). Furthermore, results show that the initial debt is rarely caused by only one of the mechatronic disciplines (Vogel-Heuser and Bi, 2021; Bi et al., 2021).

In the study of technical debt in software engineering, researchers have mainly focused on Code TD, Architecture TD, Design TD, and Test TD; however, in mechatronics, while software engineers do collaborate with other domains, *Infrastructure TD* and *Requirements TD* seem to bother them the most (Vogel-Heuser and Bi, 2021).

TD Management (TDM) in software engineering can be classified into the following eight activities: *TD identification*, *TD measurement*, *TD prioritization*, *TD prevention*, *TD monitoring*, *TD repayment*, *TD documentation*, and *TD communication* (Li et al., 2015). These can be adopted to TDM in mechatronics as well (Vogel-Heuser et al., 2015b). In software engineering, tools to manage, e.g., Code TD, are more mature and widely used, e.g., via SQALE plugin for Sonarqube (Mordal-Manet et al., 2009; Lenarduzzi et al., 2019) , ProDebt (Guzman et al., 2017), DebtFlag (Holvitie and Leppanen, 2013), Quamoco (Wagner et al., 2012), and TETRA (Kontsevoi et al., 2019). In mechatronics, however, solely some approaches exist to manage TD, e.g., by modeling TD with Causal Loop Diagrams (Dong and Vogel-Heuser, 2021a), with BPMN (Dong and Vogel-Heuser, 2021b), and other TDM approaches (Waltersdorfer et al., 2020; Biffl et al., 2019b,a). The applicability and adaptability of TDM approaches and tools are unfortunately low. It can be observed that most of the approaches are only applicable in the software of mechatronics systems, as they focus mainly on *Architectural TD* and *Code TD*. Few approaches can be applied on the meta-level in the mechatronics domain, since some management structure or ideas are universal for the TD metaphor. The main problems are the unknown TD characteristics needed to quantify metrics to identify and measure TD (Vogel-Heuser and Bi, 2021). Furthermore, TD includes more than just technical aspects, but the non-technical aspects (e.g., Social TD) have been neglected in automation and aPS for a long time (Vogel-Heuser and Rösch, 2015).

2.2. *Causes and consequences of TD*

Research into causes, symptoms, treatment, and consequences of phenomena is common in medical research or science. Moreover, causes and consequences, in terms of the TD phenomenon, are also core elements of the TD item (Avgeriou et al., 2016c; Li et al., 2015; Tom et al., 2013). As a result, researchers investigate common causes and consequences toward creating an approach for TDM. This section summarizes selected cause and consequence analyses. The most common way of TD data collection is either performing questionnaires in industry (Rios et al., 2020; Dong and Vogel-Heuser, 2018; Holvitie et al., 2018; Zazworka et al., 2013; Martini et al., 2018) and conducting expert interviews (Biffl et al., 2019a; Dong and Vogel-Heuser, 2018; Rios et al., 2019; Martini and Bosch, 2019; Runeson et al., 2012; Ollero et al., 2006). Comparing industrial questionnaires to expert interviews, correlations between causes and consequences cannot be acquired for the former, as experts solely state the frequency or impact of each incident. For the latter, causal relations can be derived by analyzing the TD incidents/items from the expert interviews.

In software engineering, Rios et al. (2019) analyzed TD causes and effects, with cross-company probabilistic cause–effect diagrams, by using classic Ishikawa/fishnet diagrams derived from questionnaires involving 72 participants. One chart was generated for each TD cause and TD effect to indicate the frequency of probability, organized by the category of cause or effect. For TD causes, the most frequently reported category is *Planning and management* (27%) followed by *Development issues* (20%) and *Methodology* (17%) with the most probable TD causes of *Deadline* (11%), *Non-adaptation of good practices* (4%), and *Lack of well-defined process* (5%), respectively. For TD effects, the most frequently reported categories are *Quality* (26%), *Planning and management* (23%), and *People* (20%), with the most probable TD effects of *Low quality* (11%), *Delivery delay* (11%), and *Team demotivation* (4%), respectively. In another study, Rios et al. (2020) reported the top 10 most cited overall TD causes and those that initiate other TD causes. In both cases, *Deadline* is the most mentioned one, followed by *Inappropriate planning*, *Lack of knowledge*, *Lack of well-defined process*, and *Non-adaptation of good practices*. Following Vogel-Heuser and Bi (2021) and Bi et al. (2021), the authors merged the apparent TD causes and asked 15 experts of multiple mechatronic disciplines to elaborate on their frequency and impact. The study (Bi et al., 2021) shows that causes differ in frequency and impact for each discipline. For mechanical engineering, for example, *Save costs*, *Lack of personnel resources*, and *Great complexity* are the most frequent. Still, for those with the highest negative impact, *Decision without specific knowledge though needed*, *Lack of personnel resources*, and *Changed boundary conditions* were stated. For software engineers working in mechatronics, both frequency and impact were rated as more critical than mechanical engineers. *Great complexity*, *Due to management decision*, and *No holistic view* are those voted to have the highest frequency, while *No holistic view*, *Great complexity*, and *Extra effort is underestimated* are those which show the most significant negative impact. Martini et al. (2018) observe in their work that TD causes can be, at the same time, a consequence, while both were presented as individual Architectural TD (ATD) core categories with no connection between each other. Martini and Bosch (2019) present their main findings for TD causes in ATD in their recent work – naming well-known TD causes, e.g., *Time pressure – deadlines with penalties*, *Uncertainty of use cases in early stages*, *Split of different budgets*, *Parallel development*, and *Lack of Knowledge*. Yet, in these studies, no connection between the cause and the effect can be analyzed due to the questionnaire-based case study design. In that case, no information on cause–effect relations within one TD incident can be obtained.

In case studies using expert interviews, researchers prefer to use cause–effect/consequence diagrams or models to elaborate on particular TD incidents selected from expert interviews. For instance, Biffl et al. (2019a) investigate TD items, causes, and effects in multidisciplinary production systems engineering. They named *Heterogeneous engineering data* one of the most significant challenges and proposed an adapted version of the Quality Function Deployment (QFD) model to visualize TD. Dong and Vogel-Heuser (2021a) propose to adapt the Causal Loop Diagrams to model industrial TD in production systems. Furthermore, Biffl et al. (2019b) and Dong and Vogel-Heuser (2018) map the cause–effect models of selected TD incidents. Using correlations between TD causes, items, and effects on different project levels, a map of exploratory case studies reflects individual correlations. Furthermore, Waltersdorfer et al. (2020) propose a TDM framework distinguishing TD causes, items, and effects.

Vogel-Heuser and Bi (2021) state that, "Understanding the cause-effect relationship is crucial, not causes and effects separately. [. . .] Causes and effects might be identical. [. . .] Depending on the "cut" in the timeline, you may encounter some of the characteristics as TD causes, symptoms, or consequences, indicating that there is a need to improve the causal chain analyses".

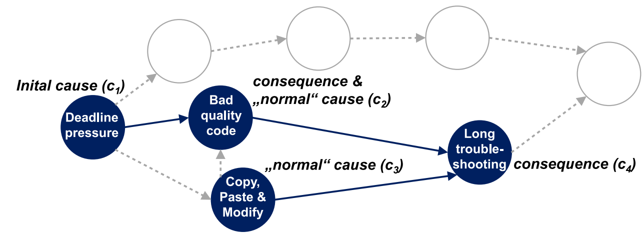

Fig. 2. Example of different understanding and graphical representation of TD causes and consequences of previous research items and proposed novel approach of TD causes/consequences and coding in this study. The initial cause coding (1) can be found in Fig. 3 in "CAUSE_ID".

We conclude that until now, researchers have distinguished between initiating causes, "normal" causes, TD items, and consequences/effects within a TD incident. As previously shown, TD incidents have a contagious character. Meanwhile, TD causes could induce TD consequences, and vice versa. Therefore, TD consequences must be treated as causes in the latter case. We propose a novel view of TD causes and TD consequences in this study (cp. Fig. 2 and Section 3.2). No correlation analysis, but only analyses on single TD characteristics have been carried out in prior research.

3. Research methodology

This section first presents the study design, including foci, e.g., the sampling strategy, the expert interviews, and the sample size. Next, we explain the problem we faced during the interview coding and elaborate on the proposed viewpoint regarding the classification of TD causes and TD consequences. Furthermore, we use an example to elaborate on our coding and the database selected. Finally, we explain the data analysis procedure we have applied.

3.1. Study design

To answer the RQs, information about confidential cases and processes in the industry regarding TD is required. Although questionnaires can elicit the frequency of TD characteristics (RQ1), questions regarding the interdependencies behind the TD incident and their initiating and affected parties (RQ2), correlations between different factors (RQ3) as well as patterns between the TD characteristics TD type, TD subtype, and TD cause (RQ4) require related information, e.g., from use cases, i.e., TD incidents. For this reason, we selected the semi-structured expert interview method (Runeson et al., 2012) to elaborate on TD in mechatronics. Only those TD incidents were included, where the experts were directly involved in, affect by and/or have been suffering from. A sample size of 47 expert interviews in 10 companies resulted in 123 industrial TD incidents that enable a picture of TD in the automation domain.

This study's design is rooted in *TDinMechatronics* (Vogel-Heuser and Bi, 2021). In this study, we focus solely on companies in the industrial automation sector as a leading sector in digitalization (Ollero et al., 2006). Same as in the previous study, we limited the study to *experts* with an educational background in mechanical, electrical, or software engineering directly involved in the product's life cycle. We qualified the experts' work experience by using the duration of time spent working in mechatronics, which has to be at least two years. Our sampling strategy was a combination of purposeful sampling and convenience sampling. We targeted companies in the industrial automation sector and aimed to interview experts with different roles and experiences. The selection was based on the availability of experts, their willingness to participate, and their relevance to our study objectives. All experts in this study participated of their free will, and all statements regarding their activities and company were treated anonymously.

We refined the interview questionnaire of (Vogel-Heuser and Bi, 2021), so the questions are more precise that the selected questions represent the research questions to be investigated to mitigate construct validity. The modified interview questionnaire[2] contains 15 questions, with three open and twelve closed questions. The four sections assessed general information on the person (Section 1, *#1–3*) and the company (*section II, #4–6*), TD related questions (Section 3, *#7–13*), and the experts' opinion on TD (*section IV, #14–15*).

The research team consisted of four researchers with different levels of experience. *Researcher A* has over 10 years of research experience in mechatronic systems and technical debt. *Researcher B* has over 40 years of research experience in interdisciplinary engineering and industrial automation. *Researcher C* has over 5 years of experience in mechanical engineering, and *Researcher D* has 5 years of experience in software engineering and interdisciplinary studies.

Each interview took approx. 30–50 min. Pre-pandemic, all interviews were conducted in in-person sessions, while we switched to online interviews during the pandemic. In the dialog, we focused solely on intentional decisions that led to TD, as the causal chain of these decisions is more clearly traceable for the investigation of cause–effect relations. In order to ensure that the questions were understood and interpreted correctly, all interviews were conducted in one-to-one sessions, with an introductory letter, graphical representation of TD, and interview questionnaire provided beforehand. As we know from the previous study, experts outside of software engineering have limited familiarity with the term TD (Vogel-Heuser and Bi, 2021). To address this issue, we evaluated the participants' prior knowledge of TD to ensure the validity of the data collected. To ensure that the experts understood the concept of TD accurately, we used the Dagstuhl 16 162 (Avgeriou et al., 2016c) definition as a reference point during the semi-structured interviews. We facilitated a dialogue between the researcher and the expert through the interview process, and the definition was presented before and during the interview to ensure that the data collected were comparable and accurate.

During the expert interviews, only those TD incidents or items were included that were directly related to the expert's involvement, experience, or suffering. The interviews were focused on intentional decisions that led to TD, as the causal chain of these decisions is more clearly traceable for investigating cause–effect relationships. All open and closed questions posed in the interview questionnaire were answered, and interviewees were allowed to expand on all answers at any time.

All interviews were recorded, transcripted, anonymized, abstracted, and checked by the expert before coding and the results' analysis were performed. The *member checking* and *peer debriefing methods* proposed by Runeson et al. (2012) have been applied to enforce the reliability of the results. The coding process involved *Researchers A* and *C* independently coding the interview transcripts. They agreed on a joint coding schema developed in the *TDinMechatronics* study. In cases of disagreement, the researchers discussed their rationale for coding the text with *Researchers B* and *D*. A consensus was reached through negotiation and clarification of the code definitions for all cases.

[2] Link to questionnaire see Appendix - Table A.1.

3.2. Challenges during interview coding

To investigate mechatronic TD characteristics, the coding of the conducted interviews constitutes one of the essential structures. Despite the coding of the demographic data (e.g., company size, experts' domain, experience in years), the coding of the TD-related information (*in Section* 3) has set up a significant challenge.

As concluded in Section 2, until now, researchers have distinguished between "initial" causes, "normal" causes, and "consequences/effects". Yet, the analysis of causes and consequences/effects shows that the list of causes is greatly similar to consequences/effects (Rios et al., 2019, 2018b). Considering the fact, that the "initial" cause is root for a "consequence/effect" (a) that could also lead to another "consequence/effect" (b) – the first "consequence/effect (a) symbolizes the "normal" cause for the subsequent "consequence/effect" (b). Fig. 2 shows an example: Due to *deadline pressure* in Project A, *bad quality code* was handed over to the customer. Later, Project B *copy-pasted and modified* the *bad quality code* of Project A, while in this case, the machine did not work at all, which led to a *long troubleshooting process*. When only considering Project A, the cause (*deadline pressure*) and the consequence (*bad quality code*) are obvious. However, while considering the TD incident in a department that contains Project A and B, we could observe that one TD incident could also have multiple TD causes. Furthermore, a consequence/effect can become a cause, namely *bad quality code* of Project A in this case in Project B. To address this phenomenon, our approach is to classify all items (causes and consequences) as numbered TD causes (cp. Fig. 2; causes = "c"). In this case, *deadline pressure* (c_1) induces bad code quality, while bad *code quality* (c_2) and *copy-paste and modify* (c_3) induce a *long troubleshooting process* (c_4). In this case, c_{min} represents the initial cause while c_{max} represents the final consequence/effect in the considered TD incident. *The long troubleshooting process* (c_4) still could be a future cause for subsequent consequences. Using this method of coding, we could elaborate on all characteristics related to TD causes without having a largely redundant list of TD consequences/effects.

3.3. Coding and database

Exemplarily, we show the coding of a TD incident in the interview dataset (cp. Table 2). It captures the Interviewee #71 (#interviewee) reported his second TD incident of the interview (#TDincident), which has five TD items (#TDitems), with each of them to be assigned to a TD type and TD subtype. Using the proposed method (cp. Section 3.2), we could code the TD causes (#TDcauses). We could see that TD item #4 has two TD causes, namely *Historically grown products* and *Most recently required for safety certifications*. For this, we have adapted the schema of Avgeriou et al. (2016c) while introducing our unique identifier "CAUSE_ID" (cp. Fig. 3). Compared to Avgeriou et al. (2016c), all our model's attributes relate to the TD Item. "TDID" enables us to evaluate all occurrences of TD characteristics within each TD incident. Derived from this coding scheme, each TD cause has its unique identifier (Uniform Resource Identifier, URI), consisting of "#interviewee-#TDincident-#TDitem-TDcause" (named "CAUSE_ID", e.g., 71-2-1-3-1). As our research questions (RQ 2–4) aim at analyzing the cause–effect relation between the TD characteristics, our database contains a large number of relationships. The graph database Neo4j is chosen to store the information for further analysis to deep-dive into each of these relationships.

Table 2
Example of a TD incident Changes can be made in late stage and without documentation from the interview data, Boundary conditions of the expert [Leading position: yes; Position: Industrial Engineering; Experience: 20+ years; Previously heard of TD: no; Company code: H2; Company size: medium; Domain: Automation] ME: mechanical engineering, EE: electrical engineering, SE: software engineering, MA: management.

#Interview	71	71	71	71	71
#TDincident	2	2	2	2	2
#TDitem	1	2	3	4	4
#TDcauses	1	1	1	1	2
TD type	Requirements TD	Documentation TD	Infrastructure TD	Process TD	Process TD
TD subtype	Late change of requirements	lack of documentation	Disregard of internal regulation	Instable process	Instable process
Cause	Lack of tolerance and guidelines	Lack of documentation guidelines	Other priorities	Historically grown products	Most recently required for safety certifications
Current state	Not solved yet	Not solved yet	Not solved yet	Not solved yet	Not solved yet
Initiating party	MA, ME, EE, SE	MA, ME, EE, SE	MA, ME, EE, SE	MA, ME, EE, SE	MA, ME, EE, SE
Disciplines involved in TD case	MA, ME, EE, SE	MA, ME, EE, SE	MA, ME, EE, SE	MA, ME, EE, SE	MA, ME, EE, SE

Table 3
Relation between research questions, methods used for answering them, survey question and section in which the information is presented.

Research question	Methods/Tools	Survey question	Section of presentation
RQ 1. What are the most frequent TD types, TD sub-types, and TD causes/consequences experienced by experts in industrial automation resulting from 368 TD items?	Semi-structured expert interview method, member checking method, peer debriefing method, interview coding, frequency analysis[a]	*#7–9*	Sections 3.1–3.3; Section 4.3.2
RQ 2. What are the interdependencies behind the TD incident and their initiating and affected parties?	Database in Neo4J and adapted schema, statical pattern analysis	*#7–9, #12*	Section 3.4; Sections 4.3, 4.3.1
RQ 3. How do company size (i), the experience of experts (ii), and their leading position (iii) relate to the reported TD characteristics?	Statical and progression pattern analysis, Chi-Square test	*#1–2, #7–9*	Section 3.2; Section 4.1
RQ 4. What patterns exist between TD types, TD sub-types, and TD causes/consequences?	Chi-Square test, PrefixSpan algorithm	*#7–9*	Sections 3.2–3.4; Sections 4.3.2–4.3.5

[a]Methods/tools were apply to all further stages.

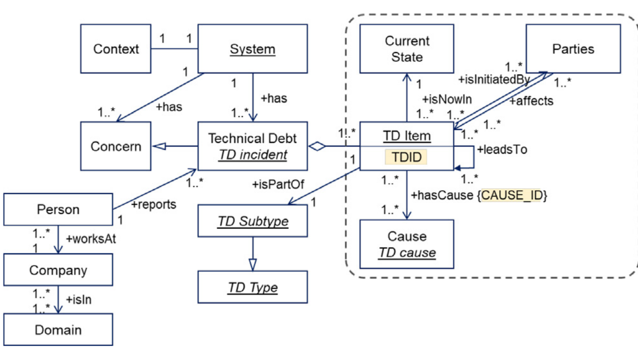

Fig. 3. Schema of the database in Neo4j, adapted from Avgeriou et al. (2016c), yellow highlights mark the IDs.

3.4. Data analysis and details of implementation

We use Python and Cypher (by Neo4j – similar to SQL for graphs) to analyze our data in this study. Python was used to automate the data import[3] to Neo4j according to the scheme proposed in Fig. 3. Furthermore, python was used for conducting data analysis to answer RQ2-4. Cypher was used to elaborate on initiating and affected parties to answer RQ 2. In addition, we used the data visualization software Tableau to display the frequency and correlations (e.g., via multi-layer bar charts and heatmaps). We conducted chi-square tests (McHugh, 2013) to elaborate on the correlations between TD type, TD subtype, and TD cause/consequence while setting $p > 0.05$. Meanwhile, we applied the PrefixSpan algorithm that enables the sequential pattern mining technique (Pei et al., 2004). It can find the frequency of sequential patterns, also in cases when in one section, a sequence is inside a larger sequence, when more than one item occurs e.g., the sequence set is "[(ac)d(cf)], [(bc)(ae)], [b], [bc]", and the pattern found by the PrefixSpan algorithm is "[c], [ca], [cb], [cc]".

4. Results of TD empirical study

In this section, we present the results of our empirical study. Starting with the descriptive statistics, we proceed to the quantitative data used to answer each RQ. To enhance the transparency and comprehensibility of the study design and the analytical process, we created Table 3 to explicitly depict the relation between the research questions, methods used for answering them, survey questions, and the sections in which they are presented.

4.1. Descriptive statistics

This study involved 47 experts from 10 companies in the automation sector. The interviewees provided 123 industrial TD incidents that contained one or more TD items. Overall, we collected a number of 368 TD items and 431 causes/consequences. There have not been two absolutely identical TD incidents. To address the uneven sample distribution of company size, the experience of the expert, and leading position, all analyses are given in a percentage value (cp. Fig. 4, table categories). The subsequent paragraphs provide an in-depth analysis of RQ 3, which aims to explore the relationship between reported TD characteristics and

[3] Structure of code for data import connecting Neo4j with Python: https://gitlab.lrz.de/TUMWAIS/td-causes-and-consequences.

factors such as *company size (i)*, *experts' experience (ii)*, and their *leading position (iii)*.

For *company size (i)*, we classify the number of employees and distinguish between large (over 250 employees) and medium (50–249 employees). Employees of large companies (70%) have reported 70 of 123 TD incidents (57%), while experts in medium-sized companies (30%) reported 53 TD incidents (43%). Hence, quantitatively, experts in medium-sized companies reported on average more TD incidents per person than in larger ones. Our approach is to discern the complexity of the TD incident by analyzing the number of TD items. For experts in large companies, 200 TD items have been reported in 70 TD incidents (2.9 TD items per TD incident). In medium-sized companies, experts reported 168 TD items in 53 TD incidents (3.1 TD items per TD incident). Therefore, the complexity here does not differ much between large and medium-sized companies.

For the *experience of the expert (ii)* in the industry, the duration of time spent working in mechatronics has to be at least two years. In addition, we classified the experts into three groups, namely 2–9 years (23%), 10–19 years (43%), and 20+ years (34%). Regarding the number of reported TD incidents, the three groups have reported 19%, 41%, and 40% of the cases, respectively. Comparing the ratios of the number of involved experts and the reported number of TD incidents, with the case per expert ratio of 0.83, 0.95, and 1.18, respectively, we could see that the greater the experts' experience, the more TD incidents they report in industrial automation.

The *leading position (iii)* refers to the level of a group manager or higher, in which the overall rate is 68% reported in 60% of TD incidents in this study. From the requirement phase until the service/retrofit, we have covered the entire product life cycle and involved experts from all three mechatronic disciplines.

A number of 74% of the experts interviewed had no prior knowledge or misinterpretation on TD before the interview. Misinterpretation could be e.g., wrong understanding of the term or the limited association of the TD term to selected TD types. In order to establish later correctness and comparability of the results, the Dagstuhl 16 162 definition by Avgeriou et al. (2016c) was introduced before and during the interview.

TD characteristics in this empirical study refer to TD type, TD subtype, and TD causes/consequences. This section elaborates on the frequency of each TD characteristic above, respectively, and compares these results to the previous studies of *TDinMechatronics* (Vogel-Heuser and Bi, 2021; Bi et al., 2021).

4.2. Frequency of TD types and TD subtypes

In order to gain a deeper insight into the correlations among TD characteristics in the field of industrial automation, the subsequent paragraph provides an in-depth analysis of the response to RQ 1, to identify the most prevalent TD types and their respective subtypes. The frequency of TD types acquired in this study is presented in the order of the life cycle of mechatronic products (cp. Fig. 4). The most frequent TD types identified were *Requirements TD* (89, 24%), *Process TD* (83, 23%), *Infrastructure TD* (44, 12%), *Test TD* (36, 10%), and *Design TD* (27, 7%). Compared to *TDinMechatronics* (Vogel-Heuser and Bi, 2021; Bi et al., 2021) , the classification was adopted when Martini et al. introduced *Process TD* (Martini et al., 2020). We endorse the separation of *Process TD* from *Infrastructure TD*, especially for the TD subtype *Sub-optimal configuration of development-related processes*. Even though we applied the new coding classification to the data, the two TD types, Process TD and Infrastructure TD, were named the two most frequent TD types. Unlike previous studies in mechatronics focusing on the industrial automation domain, *Requirements TD* and *Design TD* show similar high ratios. Yet, we record a high increase in *Test TD* and *Documentation TD*.

	Number of TD type	Company Size	Experience in years			Leading pos.
		large | medium	2-9	10-19	20+	yes | no
Requirements TD	89	13%	-1%	-15%	16%	-24%
Architectural TD	12	12%	-6%	-10%	16%	-35%
Design TD	27	48%	-19%	13%	7%	-24%
Variants TD	22	25%	-14%	-2%	16%	-9%
Versioning TD	1	70%	-23%	57%	-34%	-68%
Industrial Eng. TD	5	30%	17%	17%	-34%	12%
Manufacturing TD	2	-30%	77%	-43%	-34%	32%
Code TD	4	-30%	-23%	7%	16%	-18%
Test TD	36	10%	-9%	10%	-1%	-24%
Defect TD	18	9%	-14%	39%	-25%	-23%
Maintain./Serv. TD	4	-30%	-23%	57%	-34%	-28%
Process TD	83	15%	-5%	-2%	7%	-34%
Infrastructural TD	44	2%	9%	-11%	2%	-25%
Docu. TD	28	-7%	-5%	-29%	34%	-15%

Fig. 4. Frequency of TD types of 47 expert interviews; Percentage value is calculated by the difference of the percentage of TD type among the categories of company size (i), the experience of experts (ii), and their leading position (iii) and the percentage of expert in the respective classification.

When considering the *company size (i), experts' experience (ii)*, the analysis shows that more TD types except *Manufacturing, Code, Maintenance/Service,* and *Documentation TD* are more often reported by medium-sized companies *(i)*. Mainly *Design TD* receives the highest ratio in the comparison *(i)*. In our sample, especially experts with 10–19 years of experience report TD incidents of *Defect TD* more frequently than those with 20+ years' experience that report *Documentation TD (ii)*. It is noticeable that experts not holding a leading position report more TD incidents for all types of TD, except *Versioning TD*. *Manufacturing TD* and *Industrial Engineering TD* are the two most prominent TD types *(iii)*, yet, we only have a small sample size.

For each TD item, each TD type was further categorized into TD subtypes (cp. Fig. 5), based on the cause of the TD type, according to the classification of Li et al. (2015). Using all TD subtypes mentioned more than seven times, TD subtypes, e.g., *Instable process*, *Incomplete design specification*, *Over-engineering*, and *Insufficient performance*, were reported by experts in medium-sized companies. Meanwhile, *Insufficient documentation* is more frequently reported by experts in large companies. Given the experts' experience, experts of 20+ years emphasize TD subtypes with a reuse character, e.g., *Insufficient documentation*, *Lack of lessons learned management*, *Incomplete design specification*, and *Lack of reuse of good variant synergies*. Compared to experts with less experience (2–9 and 10–19), they instead mention TD subtypes that are addressing a more technical context, e.g., *Defects/bugs* and *Sub-optimal configuration of development-related tool.*

Subsequently, we presents the distribution of TD types and TD sub-types in correlation (cp. Fig. 5). Focusing on the TD subtypes, we could observe that few TD subtypes dominate each TD type. Especially, for the five TD types Requirements TD, Test TD, Defect TD, Process TD, and Documentation TD, there is one significant TD subtype that occurs more than all others in every case. These most frequently appeared TD subtypes are to be preferably addressed coming to TD management.

4.3. Initiating and affected parties and correlations in a TD incident

In this section, we further elaborate on the characteristics of the TD incidents and intend to uncover the parties that initiate TD incidents or are adversely affected by the effects of the TD incidents (cp. Fig.). The parties included in this study are mechanical engineering, electrical engineering, software engineering, suppliers, testing department, purchasing department,

TD Type	TD Subtype	number of
Requirements TD	Lack of requirements	24
	Blurry requirements	16
	Late change of requirements	15
	Over-engineering	7
	Insufficient internal qualification	7
	Lack of market analysis/foundation analysis	4
	Impossible to derive design from requirements	4
	Impossible to derive verification/test cases from requirements	4
	Lack of specification	3
	Wrong requirements	2
	Lack of requirements check	1
	Information asymmetry between supplier and company	1
	Disregard of safety requirements	1
Architectural TD	System-level structure dimensioning issues	6
	Complex architectural behavioral practices	3
	Violation of food architectural practices	2
	System-level structure quality issues	1
Design TD	Insufficient performance	10
	Incomplete design specification	7
	Insufficient component quality	5
	Lack of market analysis/foundation analysis	3
	Simulation model imprecise	1
	Insufficient software maturity	1
Variants TD	Lack of reuse of good variant synergies	8
	Complex variants	6
	Reuse of immature variants	4
	Immature variant architecture	2
	Redundancies are overlooks	1
	Lack of standards	1
Versions TD	Multi-version support	1
Industrial Engineering TD	Incomplete bill-of-material	3
	Unclear work instructions for supplier	1
	Assembly department not involved	1
Manufacturing TD	Wrong manufacturing process	1
	Insufficient component quality	1
Code TD	Code smells	2
	Dead code	1
	Coding violations	1
Test TD	Lack of tests	20
	Low code coverage	4
	Inadequate tests	4
	Low test coverage	3
	Lack of simulation	2
	Unclear tests	1
	Late tests	1
	Lack of test interpretation	1
Defect TD	Defects/bugs	9
	Known risk remains in series product	1
	Immature product sold	1
Maintenance/ Service TD	Required maintenance not performed	2
	Tests are not maintained	1
	disproportional invest for maintenance needed	1
Process TD	Sub-optimal configuration of dev.-rel. process	53
	Instable process	7
	Intransparent process	6
	Lack of transparency for interdisciplinary dev.	5
	Unclear process transformation for large-scale production	2
	Risk analysis is underestimated	2
	Manual programming procedure in production	2
	Too little time for process	1
	Standards not accessible for all countries	1
	Simultaneous engineering required	1
	Parallel change of multiple parties	1
	Late process planning	1
	Disregard of internal regulation	1
Infrastructure TD	Sub-optimal configuration of dev.-rel. tool	10
	Lack of lessons learned management	7
	Insufficient clarification of responsibility in interfaces	7
	Sub-optimal configuration of dev.-rel. technology	5
	Information asymmetry between supplier and company	3
	Lack of risk management	2
	Lack of reuse of good synergies	2
	Disregard of internal regulation	2
	Supplier quality varies from prototype to series item	1
	Sub-optimal operating resource	1
	Silo-thinking	1
	Obsolete guideline	1
	Lack of standards	1
	Lack of communication	1
Documentation TD	Lack of documentation	16
	Insufficient documentation	7
	Unsuitable data format	1
	Not available in required language	1
	Lack of TD documentation in ME & EE	1
	Documentation not accessible for everyone	1
	Divergence between real part and version documentation	1

TD Type	
Requirements TD	89
Architectural TD	12
Design TD	27
Variants TD	22
Versions TD	1
Industrial Engineering TD	5
Manufacturing TD	2
Code TD	4
Test TD	36
Defect TD	11
Maintenance/ Service TD	4
Process TD	83
Infrastructure TD	44
Documentation TD	28

Fig. 5. TD types and TD subtypes.

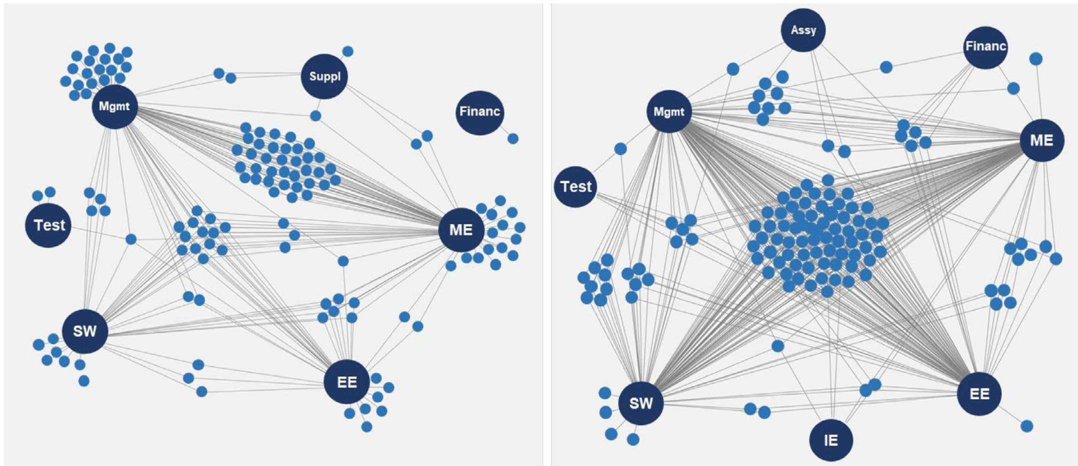

Fig. 6. Initiated (left) and affected parties (right) of the TD incidents. Each node symbolizes one TD incident, lines symbolize the connections to the correlating parties. ME: mechanical engineering, EE: electrical engineering, SW: software engineering, Suppl: supplier, Test: testing, Financ: finance/purchasing department, IE: industrial engineering, Assy: assembly, Mgmt: management.

industrial engineering, and assembly. Each node symbolizes one TD incident, while the lines between the nodes represent the connections to the involved parties. Specifically, we address RQ 2 by analyzing the interdependencies behind the TD incident and their initiating and affected parties in Section 4.3.1. Additionally, we examine RQ 4 by exploring patterns between TD types, TD sub-types, and TD causes/consequences in Sections 4.3.2–4.3.5 to provide a comprehensive understanding of the underlying relationships among the TD characteristics.

4.3.1. Initiating parties and affected parties in a TD incident

For the initiating parties of TD incidents (cp. 6, left), the most significant number of TD incidents on which we have elaborated have been initiated between the two parties' *management* (Mgmt, group leader, or higher) and the *mechanical engineering* (ME). Single parties have created many TD incidents that only hold lines to one party. Despite this, the third major group affects those initiated by multiple parties (4+), e.g., *management*, *mechanical engineering*, *electrical engineering*, and *software engineering*.

Comparing the figure of the affected parties (cp. Fig. 6, right) to the previously initiated one (cp. Fig. 6, left), we notice that far more TD incidents affect multiple (4+) disciplines. Merely six TD incidents remain that involve one single party. We could observe that the parties are more connected to each other, which means that multiple parties are forced to suffer the adverse effects of TD and are involved in examining/resolving the TD incident. Greater resources from other parties are tied up to address TD incidents from a few parties.

4.3.2. Correlation of TD types

For the sequential pattern mining, we identified that Requirements TD and Process TD are the root cause of most of the TD incidents. Fig. 7 summarizes all cause–effect relations between two TD types. The horizontal rows represent the TD type that initiates the TD incident, while the vertical columns represent the TD types which are induced and occur afterward. Considering the SUM of those initiating TD types, we could identify *Requirements TD* (88) and *Process TD* (76) as those two that are most crucial for starting the TD incidents. Meanwhile, *Requirements TD* (67), *Process TD* (62) again, and, *Infrastructural TD* (46) are those TD incidents that are induced most. If we consider three-layer sequential patterns chains of TD types, we identified three cause–effect chains that each occurred three times in our dataset, namely:

- *Requirements TD - Documentation TD - Infrastructure TD*
- *Requirements TD - Documentation TD - Process TD*
- *Requirements TD - Process TD - Design TD*

In the overall distribution (cp. Fig. 7), *Test TD*, *Maintenance TD*, *Defect TD*, and *Industrial Engineering TD* are more likely the consequence but rarely the cause, whereas for *Architectural TD*, *Version TD*, and *Requirement TD* they instead cause the TD incidents. By using the Chi-square tests and ignoring the order of initiation or cause, yet focusing on the general appearance of two TD types in one TD incident, we were able to identify the following two statistical significant dependencies ($p < 0.05$):

- *Infrastructure TD* and *Design TD* (p-value: 0.0116)
- *Documentation TD* and *Test TD* (p-value: 0.0357)

4.3.3. Correlation of TD subtypes

To further analyze the TD types, we dive deep into the TD subtypes, showing all interdependencies between the TD subtypes if they appear more than twice (cp. Fig. 8). We can observe that the *Sub-optimal configuration of development-related process* is both the most prominent TD subtype that initiates a TD incident and the most affected. Furthermore, the TD subtype *Lack of requirements* causes the subtypes *Lack of tests*, *Late change of requirements*, and *System-level structure dimensioning issues* (3). The TD types to which each TD subtype belongs can be taken from Fig. 5.

Additionally, we can observe that *Lack of documentation* can more likely induce *Blurry requirements*, *Late change of requirements*, and lead to problems of *insufficient documentation*. For the TD subtype *Lack of test*, the effects mostly stay in the same TD type. Companies or departments that lack testing most likely have more than one testing issue, even in the same TD incident, e.g., lack of unit test or lack of component test are both subsumed under the TD subtype *Lack of test*, yet address other areas of the life cycle. *Low code coverage* is also a frequent effect of the *Lack of tests* that applies to electrics and software engineering disciplines. Besides those mentioned, we found further correlations with three appearances:

- *Late change of requirements* and *Late requirements*
- *System-level dimensioning issues* and *Lack of requirements*
- *Over-Engineering* and *Insufficient performance*

Similar to Section 4.3.2, we elaborated on the three-layer sequential pattern chains of TD subtypes. Even one four-layer

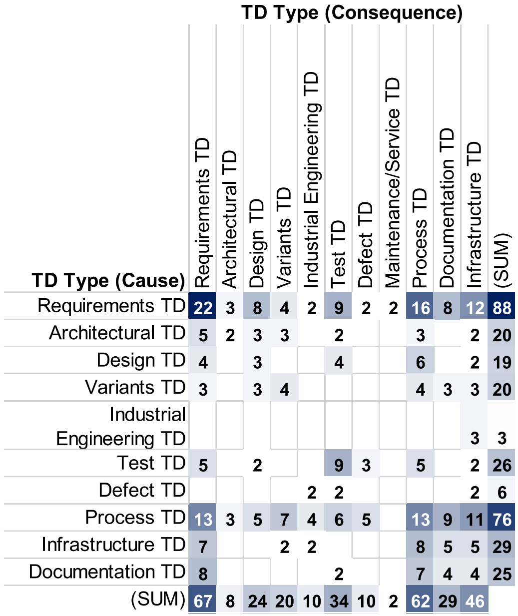

Fig. 7. Correlation of TD type, with no causing and affecting characteristics in the TD types in Versions TD, Manufacturing TD, and Code TD; No causing characteristics in Maintenance/Service TD.

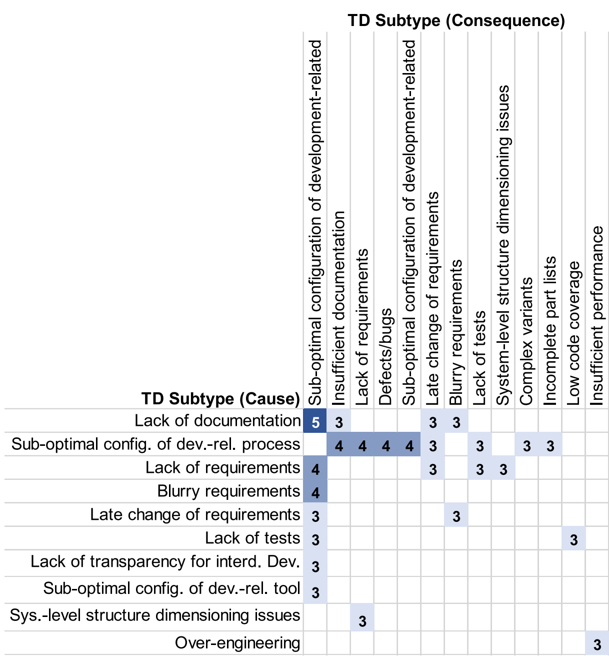

Fig. 8. Correlation of TD subtypes; n > 2.

sequential pattern has been uncovered. Here, we were able to identify the following cause–effect chains that each occurred twice in our dataset, namely:

- *Lack of requirements - Blurry requirements - Sub-optimal configuration of development-related process - Insufficient documentation*
- *Lack of requirements - Lack of tests - Low code coverage*
- *Sub-optimal configuration of development-related process - Defects/bugs - Incomplete part lists*

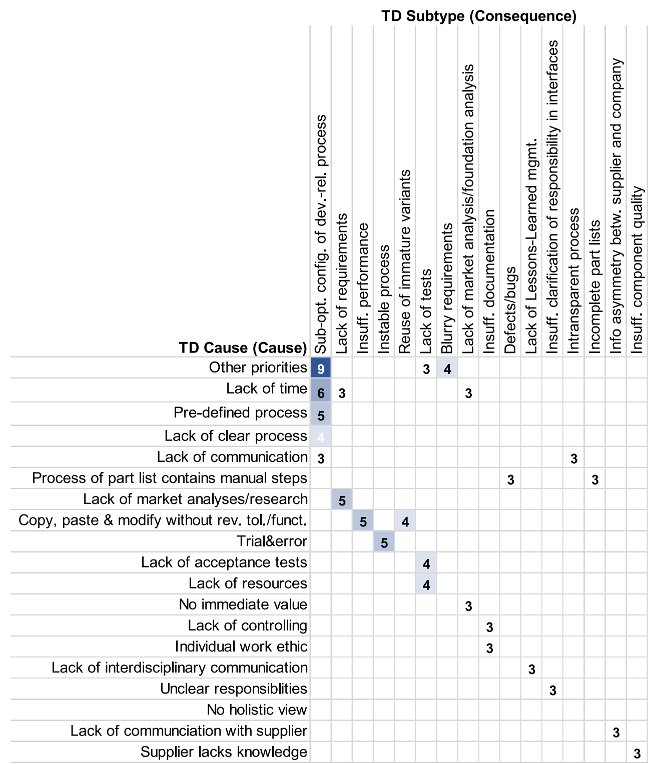

Fig. 9. Correlation of TD subtypes and TD cause/consequences; n > 2.

- *Sub-optimal configuration of development-related process - Lack of requirements - System-level structure dimensioning issues*
- *Sub-optimal configuration of development-related process - Lack of tests - Low code coverage*
- *Lack of reuse of good variant synergies - Lack of documentation - Sub-optimal configuration of development-related process*

Unfortunately, due to the low number of repetitions in the broad dataset, we were unable to reach the expected value of 5 for the Chi-square test.

4.3.4. Correlation of TD cause/consequences

In the third step, we elaborated on the correlations of TD causes/consequences. In this case, the TD cause of the first TD item symbolizes the root cause in our study. Thus, the causes of the subsequent TD items within a TD incident are also classified as a TD cause, but treated as a consequence of the initial cause. We map the distribution of all TD causes/consequences with a frequency in the correlation analysis (Pei et al., 2004) larger than two (cp. Fig. 9). Under this consideration, *Other priorities* is the one TD cause most frequently mentioned as both the root cause (16) and the consequent cause (27). Following this, the most prominent initiating TD causes for TD incidents in the automation domain are *Historically grown products* (12), *Lack of time* (9), *Lack of tolerance and guidelines* (7), and *No holistic view* (7). Those TD causes that are consequently induced are similar to the initiating ones *Historically grown products* (11), *Lack of time* (7), *No holistic view* (7), and *Most recently required for safety certifications* (6). Besides the duplicate TD causes that occur multiple times within one TD incident, there are also other correlations worth mentioning. For instance, the *Lack of tolerances and guidelines in mechanics and electrics* and *Other priorities* can lead to TD incidents (3). Another example would be that combining the *Historically grown* *products* and the *Most recent required change for safety certifications* can lead to the occurrence of TD incidents. These can serve as the foundation for the first indicators of TD prediction. By considering the three-layer sequential pattern analysis, we were able to identify the following causal chains that each occurred twice in our TD cause/consequence dataset, namely:

- *Supplier lack of knowledge - Lack of tolerance and guidelines - Lack of communication with supplier*
- *(Pre-) Defined process - Historically grown products*
- *Most recently required for safety certifications*

Due to many diverging TD causes/consequences, this analysis shows only a few repetitive patterns.

4.3.5. Correlation between TD subtypes and TD causes/consequences

In the final Section of TD correlation analysis, we elaborate on the correlations between TD subtypes and TD causes/consequences in our study.

By mapping those initiating TD subtypes and the consequent TD causes (cp. Fig. 10), this analysis shows more remarkable results comparing to the correlation of TD cause/consequences (cp. Fig. 9). According to the mapping, we may formulate the following statements based on quantitative data that supports later TD prediction, e.g.

- The lack of priorities (*Other priorities*) can cause the *Sub-optimal configuration of development-related processes* (9), *Blurry requirements* (4), *Lack of tests* (3), *Lack of reuse of good variant synergies* (2), and the *Sub-optimal configuration of development-related tools* (2)
- The *Copy, paste & modify strategy without revising the tolerances/functionality* can cause *Insufficient performance* (5), *Reuse of immature variants* (4), and *Over-engineering* (2)
- *Lack of market analysis and further research* can cause the *Lack of requirements* (5)
- The *Trial&error* approach can cause *Instable processes* (5)

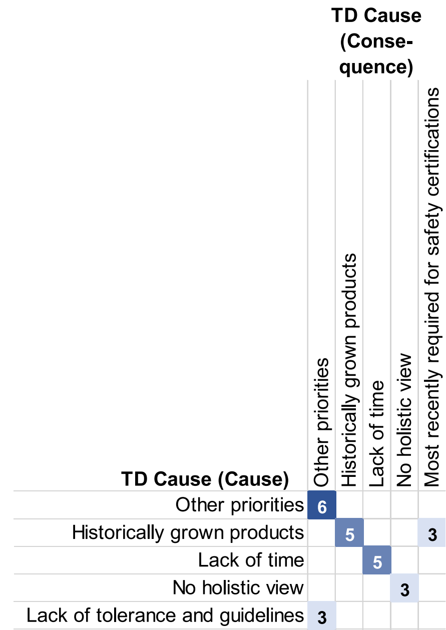

Fig. 10. Correlation of TD causes/consequences; n > 2.

5. Discussion

This section interprets the procedure and results of the study. It elaborates on their implications for researchers and practitioners who work in a mechatronic environment, and compares the study's outcomes with previous research results.

5.1. Discussion on the research approach and methodology applied

The study follows the previous research of *TDinMechatronics,* focusing on the industrial automation domain. Since there is still little understanding of TD characteristics in disciplines beyond software engineering, the focus is on the interaction with domain experts. Although the type of data collection, with one-to-one expert interviews and the subsequent systematic coding and classification, is extensive, the analyses that can be performed and the results are highly promising. It is difficult, almost impossible, to identify correlations within a TD incident when using surveys. On the one hand, experts in domains other than software engineering are not familiar with the TD metaphor (Vogel-Heuser and Bi, 2021). Even with the Dagstuhl 16 162 definition (Avgeriou et al., 2016c), some experts report that it is hard to understand where TD starts and what a TD incident can be like. Most surveys solely gather the frequency of TD characteristics, e.g., TD types, TD causes, and TD consequences (Rios et al., 2020, 2019, 2018b).

In our study design, we further faced the issue of coding TD causes and TD consequences. Previous studies show considerable overlap between the TD causes and TD consequences (Rios et al., 2020, 2019). For example, *Lack of time* can be a cause for initiating a TD incident, yet, *Lack of time* can also be a consequence due to *Late requirements*. Therefore, we suggest a type of coding that does not differentiate between causes and consequences in the coding (cp. Table 2 and Fig. 3).

We number these TD causes, while the first cause is the "root cause", and the following ones both represent the consequence of the root cause and cause of the latter TD items (cp. Fig. 2). We are aware that the "root cause" identified in this study does not necessarily represent the ultimate root cause, as this information starts where the interviewed expert gets involved. Nevertheless, this way of coding allows the analysis of causal chains that reveals yet unknown TD characteristics. We selected a graph database to ease the representation of coherent data. To analyze the correlations, we chose the existing statistical methods of the Chi-square test (McHugh, 2013) and the mining of sequential patterns of PrefixSpan (Pei et al., 2004). Using the Chi-square test method, we faced significant problems due to the large variety of TD subtypes and TD causes. For Chi-Square tests, the test constraints dictate that the value expected should be five or more in at least 80% of the cells, and no cell should have an expected value of less than one. Plenty of the analyses in this paper face this problem, wherefrom we deduce that it is not possible to show statistical significance towards correlating TD subtypes and/or TD causes based on this sample dataset.

5.2. Discussion of the study's findings

In this study, we investigated the frequency of TD characteristics *TD type*, *TD subtype,* and *TD cause*, in correlation to the three classification criteria *company size (i), experts' experience (ii),* and *leading position (iii)*. We identified Requirements TD, Process TD, and Test TD as the three most frequently occurring TD types. In this study, although 70% of experts are from large companies, the 30% of experts from medium-sized companies provided 43% of the TD incidents. From the analyses, we identified different issues for each group. For example, experts with 2–9 years of experience suffer primarily from a lack of interdisciplinary communication and the great complexity of mechatronic products *(ii)*, while experts of 20+ years of experience complain mostly about TD subtypes with a reuse character, e.g., *Insufficient documentation*, *Lack of lessons learned management, Incomplete design specification,* or *Lack of reuse of good variant synergies.* With this new information at hand, each group can be supported in terms of their foci and addressed with problem-oriented TDM approaches. The tendency is less clear when analyzing the experts' leading position *(iii)*. The unclear trends related to the experts' position can be attributed to the binary classification of management (group leader or higher) and expert. By using this classification, the tendency of, e.g., a person in the higher management (CTO/CAO/...) cannot be distinguished from a person with fewer management duties that are involved on the manufacturing/operation-level (group leader of <10 experts). This effect can be obtained with a reclassification by applying multi-level coding. A similar phenomenon can be detected in the TD subtype *Sub-optimal configuration of development-related processes (Process TD, 53)*. This is the most frequently mentioned TD subtype, with more than twice as many mentions as the subsequent *Lack of requirements (Requirements TD, 24)*. The reason is that in many cases, the lack of procedural steps is often the root of an inadequate process with missing steps. Processes exist on different levels that impact the organization or the adverse extent of TD. For example, the lack of a mandatory component testing phase differs significantly from inadequate business processes. Until now, they are subsumed under the same TD subtype and need to be further distinguished for a more informative message.

Furthermore, this study elaborated on typical patterns that reoccur and characteristics found in multiple TD incidents. For instance, we could identify a statistical correlation between *Infrastructure TD* and *Design TD* or *Documentation TD* and *Test TD*. Concerning the correlation of TD subtypes and TD causes, e.g., the *Copy, paste & modify strategy without revising the tolerances/functionality* can cause *Insufficient performance* (5), *Reuse of immature variants* (4), *and Over-engineering* (2). Or the *Trial&error* approach can cause *Instable processes* (5). Later investigations, e.g., TD prediction or TD identification, can be enabled with this information at hand. If one of the TD characteristics appears, there is a certain probability that the others follow.

Nevertheless, the large variance and low repetition of TD causes/consequences pose difficulties for the systematic detection of correlations. In this study, we generated better results while mapping the TD subtype to the TD cause to derive correlation statements. This seemingly cross-characteristic mapping does not violate definitions, as TD types are further categorized into TD subtypes based on their *cause* (Li et al., 2015). This shows that causes cannot be assigned to a distinct TD subtype or TD type, unlike TD subtypes in a fixed relation to TD types. For instance, the frequent TD causes *Lack of time* can be the cause of the TD subtype *Lack of reuse of good variant synergies (Variants TD)* or the TD subtype *Lack of requirements (Requirements TD)*. These quantitative results are the first step toward deriving TDM metrics for TD in automation.

The graph-based representation with nodes and edges is a novel way of visualizing the parties who initiate and are subsequently affected by one TD incident. One possible reason is that until now, few studies exist outside the domain of software engineering. Although the method is novel, the results are in line with insights from previous surveys. TD incidents are most likely to involve multiple parties to rework the issue than remain isolated to the parties that caused it.

5.3. Results compared to previous studies

Researchers have been investigating TD characteristics in software engineering for almost three decades. Concepts of TD in software engineering can be and are applied to TD research in other domains (Vogel-Heuser et al., 2015b; Waltersdorfer et al., 2020; Biffl et al., 2019b; Dong and Vogel-Heuser, 2018). One of the main differences between this TD causes and consequences study and previous ones is the classification of the TD causes (Dong and Vogel-Heuser, 2021a; Biffl et al., 2019b; Rios et al., 2019, 2018b). A high overlap existed between TD causes and TD consequences in software engineering and interdisciplinary research studies. To mitigate this overlap, our proposed method subsumes both items into TD causes due to their mutual correlation affecting one another.

Due to the new way of coding that enables causal chain analyses, which has not been carried out quantitatively in previous research, a comparison of frequency-featured studies can be presented. By analyzing the TD characteristics, we see that TD types at the beginning of the product life cycle have been occurring most frequently. Besker et al. (2018) note that "In [...] startups and mature organizations, there is often a peak of accumulated TD at the beginning of feature development". Likewise, in *TDinMechatronics*, those TD types that can be assigned to a life cycle phase, Requirements TD, Architectural TD, and Design TD, all at the beginning of development, are the most frequent TD types. However, in this study concerning industrial automation, Test TD is the second most frequent TD type after Requirements TD. This finding follows the research results of Ampatzoglou et al. (2016), who stated, "[...] the most frequently identified TD type is test debt". Bi et al. (2021) state that: Code TD and Versioning TD seem to occur rarely in mechatronics". This also applies to the industrial automation domain, whereas Variants TD strongly dominates Versioning TD.

Surprisingly, the TD cause *Other priorities* overtakes the previously most mentioned TD cause *Deadlines* (Rios et al., 2018b), yet both causes are nevertheless implicitly linked. This result is similar to the study of Bi et al. (2021), where *Lack Of Time/Time To Market/Time To Customer* is considered the most frequent TD cause. In addition, the TD causes of this study *Historically grown products* (3rd) and *Lack of clear process* (4th) were not highly ranked in previous studies, but have a large impact on the industrial automation domain.

Martini and Bosch (2019, 2017) uncovered the contagious character of Architectural TD. Vogel-Heuser and Bi (2021) proved this phenomenon for mechatronics disciplines. With the investigation of RQ2, this study visualizes the infectious nature of TD towards the initiating and subsequently involved parties. The reason for the differences identified may originate from the different settings of the study, or this sample of experts may have assimilated the subject of TD differently.

5.4. Threats to validity

For this empirical study, Yin's four aspects of validity (Yin, 2013) are discussed in detail in the context of the threats to validity identified that may jeopardize the validity of the results.

5.4.1. Construct validity

Construct validity refers to how selected measures represent the research questions to be investigated. Great importance was attached to the correct understanding of the study's framework and the TD-related terms. To mitigate this threat, similarly to *TDinMechatronics*, precise information in introductory slides and a flyer are given to the experts before the interview. Furthermore, specific definitions related to TD (Avgeriou et al., 2016c,a) were clarified upfront and during the interview. We revised the interview questionnaire of *TDinMechatronics* to address the RQs more accurately. The following questions, directly and indirectly, address the research questions to be investigated, namely #7–#9 addressing RQ1, #1–#2, #7–#9 addressing RQ3, #7–#11 addressing RQ4, as well as #7–#13 addressing RQ2. For the classification of the TD characteristics (TD type and TD subtype), we strictly followed the structure of Li et al. (2015). The design of the TD causes/consequences classification architecture (cp. Fig. 3) was intensely discussed and carefully revised by three experts with experience in modeling and TD to investigate RQ4. Unfortunately, due to the COVID-19 pandemic, we could not conduct all expert interviews face-to-face, yet all sessions were conducted one-to-one.

5.4.2. Internal validity

Internal validity refers to the accuracy of inferences for different factors, e.g., if one factor affects another. Similar to *TDinMechatronics*, we applied the multi-track recruiting of the experts to keep the selection of experts as objective as possible. Furthermore, we strove to interview experts from various life cycle phases from multiple backgrounds (mechanical, electrical, and software engineering). We included experts of different company sizes, years of experience, and management levels. To address the uneven distribution in the areas mentioned above, all analyses to answer RQ3 are given a percentage value.

Regarding the classification of the TD characteristics, the structure of classification proposed by Li et al. (2015) is very coarse-grained, according to the authors themselves. The three TD characteristics of TD type, TD subtype, and TD cause correlate to each other according to their definitions, as TD types are further categorized into TD subtypes based on their cause. However, this dependency does not imply a bidirectional correlation between the TD subtype and TD cause/consequence. Furthermore, the coarse-grained classification of the *Leading position* and the TD subtype *Sub-optimal configuration of development-related process* may bias the results' precision.

5.4.3. External validity

External validity refers to the extent needed to generalize the findings. A sample of 47 interviewees from the industrial automation domain allows only limited generalizability. Yet, the experts of this study show broad coverage regarding the anticipated categorization of company sizes, years of experience, and management levels. Phenomena reasoned from analysis results correspond to TD artifacts proposed in previous studies. For example, we were able to identify 10 of 11 TD types from previous literature on software development. Unfortunately, due to the low number of repetitions in the broad dataset, we were unable to reach the expected value of 5 for the Chi-square test (McHugh, 2013) for many TD characteristics. Many diverging TD cause/consequences show only few repetitive patterns. The proposed method of TD cause/consequence classification unveils the first findings of statistical correlations, e.g., for TD types, and the first ideas of correlations between other TD characteristics. Although an incomplete dataset might bias the analysis results, the proposed classification and information elaboration methods and the first insight that correlations exist between TD characteristics are generalizable contributions. In the industrial automation domain, the issues and the TD types gathered do not directly correlate to the number of TD incidents in particular experts' companies or daily work. The experts presented the issues that, in their view, were most relevant and significant.

Furthermore, it is possible that the expert's view of the TD incidents is subjective and not holistic due to the limited view of selected product life cycle phases in which they are involved. The root cause may not be known to the expert, while snowball effects might lead to consequences experts cannot foresee. To mitigate this threat, 68% of the experts hold a management position that gives them a better overview of projects and processes than specialized experts.

5.4.4. Reliability

Reliability refers to the extent that a specific researcher influences the data or the analysis. During the interviews, deliberately, no concrete example of TD incidents was given, as it might influence the experts' understanding of TD or suggest TD incidents of a distinct TD type. If a question was misunderstood, the question was rephrased and explained in an unbiased manner. All interviews were audio-recorded, which improves the transparency and comprehensibility of discussions. Consequently, all recordings were transcribed into text. Although the same researcher conducted all interviews, we applied the member checking method (Runeson et al., 2012), so all experts had the opportunity to revise the interview transcript. If the information captured did not represent their intention, they were allowed to correct and adjust the statements. Only minor changes were handed in and subsequently adopted. Thus, the transcripts represent the original opinion of the expert. The coding and classification were carried out by applying the peer debriefing method (Runeson et al., 2012). A group of independent researchers was involved in the discussion and analysis.

6. Conclusion and outlook

This section concludes the study's findings, highlights the study's potential, and presents the future outlook.

6.1. Conclusion

This study is the first in-depth study quantifying TD causes/consequences in the multidisciplinary industrial automation domain. Until now, although TD has been a research item for almost three decades, no larger-scaled expert interviews were conducted and analyzed concerning causal chains within TD incidents. This study proposes a novel modeling and coding method for TD causes to mitigate the large number of duplicates created when classifying TD causes and TD consequences separately. This concept is not limited to the automation domain and can be applied to software engineering. Forty-seven semi-structured expert interviews in ten companies were conducted, revealing 123 mechatronic TD incidents that included 368 TD items.

Firstly, we elaborated on the frequency of TD types, TD subtypes, and TD cause/consequences in all TD incidents gathered to answer RQ1. We adopted the divisions proposed by Rindell et al. (2019) for Process TD to classify the TD types of Infrastructural TD (Li et al., 2015). Requirements TD, Process TD, and Test TD are the most common TD types in companies, whereas Test TD occurs surprisingly often in industrial automation. Build TD, Versioning TD, Manufacturing TD, Code TD, and Maintenance/Service TD feature the least. Furthermore, compared to the previous study of *TDinMechatronics* (including other domains, e.g., automotive industry, food industry), Defect TD and Documentation TD appeared at an above average rate. *Sub-optimal configuration of development-related process* is the most frequent TD subtype, yet, it lacks an indicator for processes at the management/process level at which the TD occurs. The TD causes/consequences *Other priorities* and *Lack of time* seem to be omnipresent causes for TD. However, further ones, e.g., *Historically grown products* (3rd), *Lack of market analysis and further research* (5th), and *Copy, paste & modify without revising the tolerances/functionalities* (6th), are eminent in the industrial automation domain.

Secondly, to answer RQ2, we visualize the contagious character of TD in the industrial automation domain by mapping the initiating parties and the ones subsequently affected, utilizing the graph representation of Neo4J. This study exposes the contagious nature of TD and further visualizes it among all TD incidents. Moreover, it verifies that TD seldom remains isolated at the point of the initiating parties but spreads throughout, subsequently becoming involved in the rework and repayment of TD.

Then, we related the frequency of the selected TD characteristics to the company size, the experts' experience, and position to answer RQ3. We identified different foci for these groups. For example, medium-sized companies are more likely to suffer from *Instable processes* and *Incomplete design specifications,* while large companies report more issues regarding *Insufficient documentation*. In addition, experts with 20+ years' experience are more likely to report TD subtypes that are reuse-based, e.g., relating to *Lack of reuse of good synergies*, *Instable processes*, and *Lack of lessons learned management*. In comparison, younger experts (2–9 years) complain about *Sub-optimal configuration of development-oriented tool* and *Insufficient clarification of responsibilities in interfaces*. Different focus groups are affected by different TD incidents. Therefore, for TD management, we need to address the parties adequately according to their particular problems.

Finally, we investigated existing TD patterns by applying the quantitative methods of the Chi-square test and PrefixSpan. For each TD characteristic, we were able to identify correlations. However, the Chi-square test was only applied to the TD type correlation analysis due to the large variety of TD subtypes and TD causes/consequences. Using the PrefixSpan algorithm, the correlation between TD subtypes and TD causes/consequences showed the most promising results. Yet, we could only identify 2–3 repetitive patterns among all the data.

6.2. Outlook

For future research, the state of knowledge and a transparent view of challenges in interdisciplinary engineering regarding TD management needs to be determined. Once the coding of cause–effect chains is unlocked, future research on TD can deep-dive into unveiling interdependencies between TD characteristics. Traceable cause–effect relationships between organizations, projects, discipline-related data, and TD characteristics can help researchers and practitioners understand both statistic and progression patterns.

Due to the uneven distribution of the interviews and the wide range of dispersion, more distinct expert interviews towards the experts' background would not necessarily achieve a more explicit understanding and distinct prediction of cause–effect-relations of technical debt in industrial automation. However, the result of this study can be used as a qualitative input to the investigation of TD patterns and creating TDM metrics.

The understanding, acceptance, and communication of TDM through selecting and implementing an appropriate interaction concept for TD can support the identification and evaluation of causalities/correlations, and validate the identified patterns and metrics. Early detection of possible TD in organizations through analyses of TD patterns and metrics as a preliminary stage of TD prevention can positively affect the system's general health. Further knowledge exchange, e.g., cross-company and cross-industry lessons from TD incidents must be enabled. The transparent presentation of the interrelationships, adverse extent, and risks caused by TD is the first step for introducing TDM in interdisciplinary engineering.

TDM approaches/methods/tools in mechatronics cannot be limited to those that solely address the management of certain TD aspects, e.g., "identification of Requirements TD" or "prevention of Architectural TD". Yet, the terms can further include assistance functionalities that support the selection and application of adequate existing tools that address TD as a side effect.

CRediT authorship contribution statement

Fandi Bi: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Visualization, Writing – original draft, Writing – review & editing, Software, Visualization, Project Administration, Funding acquisition. **Birgit Vogel-Heuser:** Conceptualization, Investigation, Methodology, Resources, Supervision, Validation, Writing – review & editing, Project Administration, Funding acquisition. **Ziyi Huang:** Visualization, Data curation, Formal analysis, Methodology, Software. **Felix Ocker:** Conceptualization, Methodology, Validation, 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.

Data availability

The data that has been used is confidential

Acknowledgments

The authors wish to thank the participating experts and companies for their support with informative industrial issues and their trust and openness during the study. This work was supported in part by the German Research Foundation (Deutsche Forschungsgemeinschaft) via the project TDebituM (grant no. VO937/39-1).

Appendix

Interview questionnaire (see Table A.1).

Table A.1
Interview questionnaire – answer type, coding, result of each question and result presentation in the paper.

#	Questions	Answer type	Coding
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
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+
3	Have you heard about "TD" before?	yes/no/open answer	yes/no/yes, but wrong interpretation
II. General descriptive information about the company			
4	Briefly explain your project process/development process.	Open answer	No coding

(*continued on next page*)

Table A.1 (*continued*).

#	Questions	Answer type	Coding
5	How do you communicate within the interdisciplinary team?	Open answer	No coding
6	Which tools do you use in your daily business for what (communication, V/V, ticket system, MES, ERP, SAP)? What information are available there? In which form are the information available (pdf, excel, etc.)?	Open answer	No coding
III. TD-related questions - Cause effect graph (for each TD incident)			
7	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
		Open answer - TD item	TD type
		Open answer - TD item	TD subtype
8	What is the root cause for the suboptimal decision to be made?	Open answer - TD cause (root cause)	TD cause c_0
9	What are the consequences of the TD item/TD incident?	Open answer - TD cause/consequence	TD cause c_n
10	What are the measures to resolve the TD item/incident?	Open answer	No coding
11	What is the current state of the TD item/incident?	Open answer	No coding
12	Which disciplines within the company were involved?	Initiating discipline/party	ME/EE/SE/other parties
		Affected discipline/party	ME/EE/SE/other parties/none
13	Are there any control mechanisms to the specific TD incident/item in the company and have they been complied with?	Open answer	No coding
IV. Final questions			
14	Is there an awareness of TD in your organization, the conscious decision of a suboptimal solution? Briefly explain.	Open answer	No coding
15	Where do you see the biggest challenge regarding TD repayment in your company?	Open answer	No coding

References

Ampatzoglou, A., et al., 2016. The perception of technical debt in the embedded systems domain: An industrial case study. In: 2016 IEEE 8th International Workshop on MTD. Raleigh, NC, USA, pp. 9–16.

Avgeriou, P., Ernst, N.A., Nord, R.L., Kruchten, P., 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, P., Kruchten, P., Nord, R.L., Ozkaya, I., Seaman, C., 2016b. Reducing friction in software development. IEEE Softw. 33 (1), 66–73. http://dx.doi.org/10.1109/MS.2016.13.

Avgeriou, P., Kruchten, P., Ozkaya, I., Seaman, C., 2016c. Managing technical debt in software engineering (dagstuhl seminar 16162). http://dx.doi.org/10.4230/DagRep.6.4.110.

Besker, T., Martini, A., Bosch, J., 2018. Technical debt cripples software developer productivity. In: Proceedings of the 2018 International Conference on Technical Debt - TechDebt '18. Gothenburg, Sweden, pp. 105–114.

Bi, F., Vogel-Heuser, B., Xu, L., 2021. Frequency and impact of technical debt characteristics in companies producing mechatronic products. In: 2021 IEEE/ACM International Conference on Technical Debt (TechDebt). Madrid, Spain, pp. 26–35.

Biffl, S., et al., 2019a. Software engineering risks from technical debt in the representation of product/ion knowledge. In: Proceedings of the 31st International Conference on Software Engineering and Knowledge Engineering. pp. 693–700.

Biffl, S., et al., 2019b. Technical debt analysis in parallel multi-disciplinary systems engineering. In: 2019 45th Euromicro Conference on Software Engineering and Advanced Applications. SEAA, Kallithea-Chalkidiki, Greece, pp. 342–346.

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

Dong, Q.H., Vogel-Heuser, B., 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.

Dong, Q.H., Vogel-Heuser, B., 2021a. Modelling industrial technical compromises in production systems with causal loop diagrams. IFAC-PapersOnLine 54 (4), 212–219. http://dx.doi.org/10.1016/j.ifacol.2021.10.036.

Dong, Q.H., Vogel-Heuser, B., 2021b. Modelling technical compromises in electronics manufacturing with BPMN+TD – an industrial use case. IFAC-PapersOnLine 54 (1), 912–917. http://dx.doi.org/10.1016/j.ifacol.2021.08.108.

Guzman, L., Vollmer, A.M., Ciolkowski, M., Gillmann, M., 2017. Formative evaluation of a tool for managing software quality. In: 2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. ESEM, Toronto, ON, Canada, pp. 297–306.

Holvitie, J., Leppanen, V., 2013. DebtFlag: Technical debt management with a development environment integrated tool. In: 2013 4th International Workshop on MTD. San Francisco, CA, USA, pp. 20–27.

Holvitie, J., et al., 2018. Technical debt and agile software development practices and processes: An industry practitioner survey. Inf. Softw. Technol. 96, 141–160. http://dx.doi.org/10.1016/j.infsof.2017.11.015.

Kontsevoi, B., Soroka, E., Terekhov, S., 2019. TETRA, as a set of techniques and tools for calculating technical debt principal and interest. In: 2019 IEEE/ACM International Conference on Technical Debt (TechDebt). Montreal, QC, Canada, pp. 64–65.

Lenarduzzi, V., Martini, A., Taibi, D., Tamburri, D.A., 2019. Towards surgically-precise technical debt estimation: early results and research roadmap. In: Proceedings of the 3rd ACM SIGSOFT International Workshop on Machine Learning Techniques for Software Quality Evaluation - MaLTeSQuE 2019. Tallinn, Estonia, pp. 37–42.

Li, Z., Avgeriou, Liang, P., 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, A., Besker, T., Bosch, J., 2018. Technical debt tracking: Current state of practice. Sci. Comput. Program. 163, 42–61. http://dx.doi.org/10.1016/j.scico.2018.03.007.

Martini, A., Besker, T., Bosch, J., 2020. Process debt: a first exploration. In: 2020 27th Asia-Pacific Software Engineering Conference. APSEC, Singapore, Singapore, pp. 316–325.

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

Martini, A., Bosch, J., 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.

Martini, A., Bosch, J., 2019. Architectural technical debt in embedded systems. In: Kenett, R.S., Swarz, R.S., Zonnenshain, A. (Eds.), Systems Engineering in the Fourth Industrial Revolution. Wiley, pp. 77–103.

McHugh, M.L., 2013. The chi-square test of independence. Biochem. Med. 23 (2), 143–149. http://dx.doi.org/10.11613/BM.2013.018.

Mordal-Manet, K., et al., 2009. The squale model — A practice-based industrial quality model. In: 2009 IEEE International Conference on Software Maintenance. Edmonton, AB, Canada, pp. 531–534.

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

Nord, R.L., Ozkaya, I., Kruchten, P., Gonzalez-Rojas, M., 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. Helsinki, Finland, pp. 91–100.

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.

Pei, J., et al., 2004. Mining sequential patterns by pattern-growth: the PrefixSpan approach. IEEE Trans. Knowl. Data Eng. 16 (11), 1424–1440. http://dx.doi.org/10.1109/TKDE.2004.77.

Rindell, K., Bernsmed, K., Jaatun, M.G., 2019. Managing security in software. In: Proceedings of the 14th International Conference on Availability, Reliability and Security. Canterbury CA United Kingdom, pp. 1–8.

Rios, N., Oliveira Spinola, R., Mendonca, M., Seaman, C., 2019. Supporting analysis of technical debt causes and effects with cross-company probabilistic cause-effect diagrams. In: 2019 IEEE/ACM International Conference on Technical Debt (TechDebt). Montreal, QC, Canada, pp. 3–12.

Rios, N., Oliveira Spinola, R., de Mendonca Neto, M.G., Seaman, C., 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, Prague, pp. 429–436.

Rios, N., Spínola, R.O., Mendonça, M., Seaman, C., 2018b. The most common causes and effects of technical debt. In: Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement - ESEM '18. Oulu, Finland, pp. 1–10.

Rios, N., Spínola, R.O., Mendonça, M., Seaman, C., 2020. The practitioners' point of view on the concept of technical debt and its causes and consequences: a design for a global family of industrial surveys and its first results from Brazil. Empir. Softw. Eng. 25 (5), 3216–3287. http://dx.doi.org/10.1007/s10664-020-09832-9.

Runeson, Per, Höst, Martin, Rainer, Austen, Regnell, Björn (Eds.), 2012. Case Study Research in Software Engineering: Guidelines and Examples, first ed. Wiley, Hoboken, N.J.

Tom, E., Aurum, A., Vidgen, R., 2013. An exploration of technical debt. J. Syst. Softw. 86 (6), 1498–1516. http://dx.doi.org/10.1016/j.jss.2012.12.052.

Vogel-Heuser, B., Bi, F., 2021. Interdisciplinary effects of technical debt in companies with mechatronic products — a qualitative study. J. Syst. Softw. 171, 110809. http://dx.doi.org/10.1016/j.jss.2020.110809.

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, B., Rösch, S., 2015. Applicability of technical debt as a concept to understand obstacles for evolution of automated production systems. In: 2015 SMC, Kowloon Tong, Hong Kong. pp. 127–132.

Vogel-Heuser, B., Rösch, S., Martini, A., Tichy, M., 2015b. Technical debt in automated production systems. In: 2015 IEEE 7th International Workshop on MTD. Bremen, Germany, pp. 49–52.

Wagner, S., et al., 2012. The Quamoco product quality modelling and assessment approach. In: 2012 34th International Conference on Software Engineering. ICSE, Zurich, Switzerland, pp. 1133–1142.

Waltersdorfer, L., Rinker, F., Kathrein, L., Biffl, S., 2020. Experiences with technical debt and management strategies in production systems engineering. In: Proceedings of the 3rd International Conference on Technical Debt. Seoul Republic of Korea, pp. 41–50.

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

Zazworka, N., Spínola, R.O., Vetro', A., Shull, F., Seaman, C., 2013. A case study on effectively identifying technical debt. In: Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering - EASE '13. Porto de Galinhas, Brazil, p. 42.

Fandi Bi, M. Sc received the Master of Science degree in mechanical engineering in 2019 and is currently pursuing her Ph.D. degree at the Institute of Automation and Information Systems at the Technical University of Munich. Her research interests primarily revolve around identifying and measuring technical debt issues in mechatronic systems. She was awarded the Verein Deutscher Ingenieure (VDI) Prize 2019 in the category of best master's thesis on the topic of "Interdisciplinary effects of technical debt in companies with mechatronic products", as well as the TUM Faculty of Mechanical Engineering WITTENSTEIN Prize 2020 in the category of best master's thesis.

Birgit Vogel-Heuser, Prof. Dr.-Ing.*(Fellow, IEEE)* is a full Professor and Director of the Institute of Automation and Information Systems at the Technical University of Munich. Her main research interests are systems and software engineering, and modeling of distributed and reliable embedded systems. Vogel-Heuser is a member of the German Academy of Science and Engineering (acatech), served as General Chair (2018) and Keynote Speaker (2022) at the IEEE International Conference on Automation Science and Engineering (CASE), Associate Editor (2016–2019) and Senior Editor (since 2019) of IEEE Transactions on Automation Science and Engineering (T-ASE), and Distinguished Lecturer of two IEEE Robotics and Automation Society (RAS) Technical Committees (since 2021).

Ziyi Huang, M.Sc., received the Master of Science degree in mechanical engineering in 2023 at the Technical University of Munich. Her research interests lie in automation, correlation analysis, and technical debt management.

Felix Ocker, Dr.-Ing., received his doctoral degree in engineering from the Institute of Automation and Information Systems at the Technical University of Munich in 2022. His research focuses on knowledge formalization and inconsistency management in interdisciplinary engineering.