The Journal of Systems and Software 167 (2020) 110586

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

# The influence of Technical Debt on software developer morale

Terese Besker[a,*], Hadi Ghanbari[b], Antonio Martini[c], Jan Bosch[a]

[a] Chalmers University of Technology, Hörselgången 4, 417 56 Gothenburg, Sweden
[b] Aalto University, Ekonominaukio 1, 02510, Espoo, Finland
[c] University of Oslo, Department of Informatics, Gaustadalléen 23B, 0373 Oslo, Norway

ARTICLE INFO

*Article history:*
Received 11 October 2019
Revised 11 March 2020
Accepted 20 March 2020
Available online 29 April 2020

*Keywords:*
Software development
Technical debt
Developer morale
Software productivity
Technical debt management
Human factors

ABSTRACT

*Context:* Previous research in the Technical Debt (TD) field has mainly focused on the technical and economic aspects, while its human aspect has received minimal attention.

*Objective:* This paper aims to understand how software developers' morale is influenced by TD and how their morale is influenced by TD management activities. Furthermore, this study correlates the morale with the amount of wastage of time due to TD.

*Method:* Firstly, we conducted 15 interviews with professionals, and, secondly, these data were complemented with a survey. Thirdly, we collected 473 data points from 43 developers reporting their amount of wasted time. The collected data were analyzed using both quantitative and qualitative techniques, including thematic and statistical analysis.

*Results:* Our results show that the occurrence of TD is associated with a lack of progress and waste of time. This might have a negative influence on developers' morale. Further, management of TD seems to have a positive influence on developers' morale.

*Conclusions:* The results highlight the effects TD has on practitioners' software work. This study presents results indicating that software suffering from TD reduces developers' morale and thereby also their productivity. However, our results also indicate that TD management increases developers' morale and developer productivity.



## 1. Introduction

In recent years, there has been an increasing interest in technical debt (TD) within the software engineering discipline. The majority of previous studies focused on investigating the technical, financial, and organizational aspects of TD (Ampatzoglou et al., 2016), (Li et al., 2015), (Besker et al., 2018). However, until now, the literature has paid minimal attention to the human, and social aspects of TD, including the role of developers in the occurrence of TD as well as its consequences for them. More recently, several studies have suggested that TD has a negative influence on developers' emotions and affects (Yli-Huumo et al., 2014), (Codabux and Williams, 2013), (Spinola et al., 2013) and their morale (Tom et al., 2013), (Fernández-Sánchez et al., 2015), where morale can be explained as a multidimensional concept that "*subsumes confidence, optimism, enthusiasm, and loyalty as well as a sense of common purpose*" (Peterson et al., 2008). However, none of these previous studies specifically focuses on *empirically* investigating and explaining the relationship between TD, the developers' morale, and software developer productivity. Since no known research has focused on exploring these relationships empirically, this study seeks to obtain data from practitioners who experience TD on a daily basis, in order to expand the empirical findings in this area.

In a study by Tom et al., (2013), the authors suggest that morale, alongside quality, productivity, and project risk, are the four main areas that are negatively influenced by the occurrence of TD. The authors claim that, since the occurrence of TD reduces software quality, developers must spend more time and effort to address quality issues in the future. This, in turn, will decrease developers' productivity and maintenance costs in the long term (Tom et al., 2013), (Fernández-Sánchez et al., 2015). On the other hand, some studies suggest that developers do not feel comfortable about taking on TD (Yli-Huumo et al., 2014), (Codabux and Williams, 2013), and it lowers their motivation (Spinola et al., 2013).

Software development is a sociotechnical phenomenon (McLeod and Doolin, 2012), and therefore, its success depends on

* Corresponding author.
*E-mail addresses:* besker@chalmers.se (T. Besker), hadi.ghanbari@aalto.fi (H. Ghanbari), antonio.martini@ifi.uio.no (A. Martini).

https://doi.org/10.1016/j.jss.2020.110586
0164-1212/

both its social and technical aspects (Feldt et al., 2010). The ways in which today's software developers work require a comprehensive understanding of their feelings, perceptions, motivations, and identification with their tasks in their respective project environments (Fagerholm et al., 2015). Recent studies have shown that positive affective states, such as happiness, satisfaction, and motivation, increase software developers' productivity and software quality (Feldt et al., 2010), (Graziotin and Abrahamsson, 2014). On the other hand, based on the results of previous studies, mainly from management science (Abbott, 2003), (Stowe, 2009) and more recently in software engineering (Damian and Chisan, 2006), (Fairley and Willshire, 2005), (Foulds and West, 2007), (Hall et al., 2007), developers' morale and their productivity seem to correlate. However, to the best of our knowledge, there is a lack of studies focusing on investigating the relationship between TD and morale and productivity using empirical data.

Considering the large number of previous studies which argue that software firms take on TD to boost their productivity and obtain added business value (Tom et al., 2013), it becomes apparent that minimal attention has been paid to the human and social aspects of TD (Becker et al., 2017), and that there is a need for empirical studies (Cunningham, 1992) to explore and explain the effects of TD on developers' morale and to correlate it with developer productivity.

The goal of this study is, therefore, to understand how software developers' morale is influenced by TD present in the software they are developing and also to understand how their morale is influenced by TD management activities. Furthermore, in order to understand if their morale affects their work productivity, this study explores associations between morale with the amount of wastage of working time due to experiencing TD.

Note that the occurrence of TD and its produced time waste are two separate concepts: companies can have TD with a low "interest rate" (the term used to characterize the negative impact of TD). In such case, the presence of TD does not produce time waste but can still influence morale. On the other hand, TD can produce time waste, which may be the reason why TD affects morale. For these reasons, we study the concepts separately and we have two separate research questions (RQ1 and RQ3).

The term developer morale will be examined by surveying developers to indicate on a Likert scale their opinion about the impact of TD on three different dimensions of morale. The three different dimensions are: a) *Affective antecedents* such as focusing on developers' moods, feelings, emotions, and attitudes, and b) *Future/goal antecedents* with a focus on the developers' goals for the future and, finally, c) *Interpersonal antecedents* addressing the relationships and communication between developers. These dimensions are described in detail in Section 2.2.

Based on this goal, this study will examine the following research questions (RQ):

- **RQ1:** Does the occurrence of TD influence developers' morale?
- **RQ2:** Does the management of TD influence developers' morale?
- **RQ3:** Does wasted time (due to TD) correlate with morale?
  - **RQ3.1:** Does wasted time correlate with *TD occurrence* dimensions of morale?
  - **RQ3.2:** Does wasted time correlate with *TD Management* dimensions of morale?

Our study has several novel contributions to software engineering research and practice. First, our study specifically concentrates on investigating the influence of TD on developers' morale. Our findings are encouraging since they clarify the impacts of TD on different dimensions of morale and illustrates its relationship with developer productivity. Especially by indicating that developers consider TD and its management as important factors influencing progress and future development activities, this study encourages software firms to consider the human and organizational consequences of TD more seriously. Additionally, since previous studies have indicated the link between morale and developers' productivity, our study investigates this relation empirically. In future research, this approach will enable enhanced exploration and measurement of software developers' morale in different contexts.

An earlier paper at the 11th International Symposium on Empirical Engineering and Measurement (ESEM) (Ghanbari et al., 2017) presented part of the results reported in this study. However, this manuscript extends that previously published study significantly by triangulating the previous data collection and adding more data and, moreover, by adding more analysis. The novelty of this study's approach compared to the previous study lies in the morale focus of this paper, where we correlate the waste of time with three different dimensions of morale together with seven additional interviews and additional survey questions related to the morale perspective of TD.

The data representing the wasted time are based on reported data from developers who were asked to individually keep track of the time they wasted due to experiencing TD, on a daily basis. They were thereafter asked to themselves report this time to us, twice a week in a survey. Meaning that the data are based on the participants' own tracking and calculations.

The original study was exploratory, and based on our initial results, we proposed a set of propositions. In this study, using those propositions as a starting point, we conducted an additional cycle of research to investigate further the validity of those propositions and also to explain the logic behind the correlation of TD, morale, and productivity. In order to investigate if developers' morale affects their productivity, we have added one additional research question (RQ3), where we correlate the reported amount of working time wasted due to experiencing TD (as a proxy of productivity) with three different dimensions of morale and discuss them in detail. The related Research section has been extended to be broader and more carefully cover additional related research publications. We have extended the data collection by adding a longitudinal study collecting additional quantitative data in order to understand how much time developers report as being wasted due to experiencing TD over a longer period of time. We also augment our previous study with seven additional interviews. We believe that this additional context extends and strengthens the results derived in the original study.

This paper is structured as follows: Section 2 presents the theoretical background of the study. In Section 3, we describe the research methodology. Section 4 presents the research results. Sections 5 and 6 discuss the findings and threats to the validity of the study, respectively. Finally, Section 7 concludes the study.

## 2. Theoretical background

In this section, we discuss the background of the study in terms of TD, morale, the relationship between TD and morale, and the relationship between TD and productivity, and finally, the relationship between morale and productivity.

### 2.1. Technical debt

In recent years, TD has been widely studied in the software engineering literature (Li et al., 2015), (Cunningham, 1992), (Brown et al., 2010). TD is composed of a debt, which is a suboptimal technical solution that leads to a short-term benefit as well as to the future payment of interest, which is the extra cost due to the presence of TD (e.g., slow feature development or low quality) (Besker et al., 2018). The principal is regarded as the cost of refactoring TD. Although accumulated TD might result in a gain

between the short-term benefit and the interest paid, in many cases, documented in the literature and software projects, the interest might largely surpass the gain by, for example, leading to development crises (Martini et al., 2015).

There are several kinds of TD, such as Architecture TD, Testing TD, and Source Code TD (Li et al., 2015), depending on where the sub-optimality has occurred, what artifact, and what level of abstraction. In recent years, there has been an increasing interest among software engineering researchers in providing novel solutions enabling software developers to manage TD. Management of TD consists of a set of activities that enable software development teams to both prevent the occurrence of TD and deal with existing TD and its unwanted consequences(Li et al., 2015). It is worth noting that, in their systematic mapping study, Li et al. (2015) classify TD management activities into identification, measurement, prioritization, prevention, monitoring, repayment, documentation and representation, and communication. However, discussing each of these activities is out of the scope of this study.

### 2.2. Human factors related to technical debt

Today's software development work is, to a great extent, a human-based activity that depends on several different human aspects (Fagerholm et al., 2015) whereby its success is dependent on financial, social, physical, and emotional factors (Ralph and Kelly, 2014).

The success of software development projects is dependent on the practitioners working with the software. For instance, Beecham et al. (2008) states that developers' motivation is an important issue, and that lack of motivation is one frequent cause of software development project failure (Beecham et al., 2008). Furthermore, Verner et al. (2014) state that motivation is considered to be the single largest factor in developer productivity.

Moreover, several researchers state that, specifically, TD has a negative impact on the practitioners' daily work with software that suffers from experiencing TD. Tom et al. (2013) state that it is likely that developers find the effects of TD frustrating in the long term, and Laribee (2009) explains that economic downside, there's a real psychological cost to technical debt that causes both frustration and a sense of helplessness to the developers and also Vogel-Heuser and Neumann (2017) state that human factors can be a reason for TD.

Further, software engineering success is described by Keyes (2011) to increasingly dependent on the well-being of developers' communities, and Tamburri et al. (2013) highlight that a suboptimal development community can have a significant impact on the software by causing unforeseen project cost. The authors refer to this suboptimality as social debt. Alfayez et al. (2018) reveal that the skills and habits of developers have an impact on the software where e.g., developer commit frequency and seniority have a significant negative relationship with the TD introduced by the developer, meanwhile, Salamea and Farré (2019) conclude that communication skills have barely any impact on TD.

### 2.3. Morale

Morale, as a research topic, has originally been of interest within the military discipline. However, after the Second world war, it has received increasing attention from other disciplines such as organizational sciences, management, education, and healthcare (Hardy, 2010). Despite its vast literature, morale lacks a coherent and precise definition, which increases the risk of researchers measuring other concepts instead of morale. For example, in an extensive literature review, Hardy (2010) shows that several concepts such as satisfaction, motivation, and happiness, have been used interchangeably to discuss morale. Similarly, França et al. (2011), describe that the terms morale, inspiration, enthusiasm, and motivation are popular synonyms even if they potentially have distinct actual meanings in the field of psychology. However, by conducting a longitudinal empirical study, Hardy 2010) clarifies that morale is different from these concepts. This study will only focus on morale as a human factor, where we use the definition of morale provided by Pterson et al. (2008) as "*a cognitive, emotional, and motivational stance toward the goals and tasks of a group. It subsumes confidence, optimism, enthusiasm, and loyalty as well as a sense of common purpose.*" As can be understood from this definition, morale is a multidimensional concept consisting of different affective, interpersonal, and motivational dimensions.

Previous studies have used different qualitative and quantitative methods for measuring morale, including direct (e.g., (McConnell, 2010), (Peters, 2014)), and indirect measurement methods (e.g., (Martini et al., 2015)). In its simplest way, morale has been widely measured directly by using single-scale measures in which individuals are asked to rate their level of morale (McConnell, 2010). However, using this approach becomes problematic since morale is a subjective phenomenon, and therefore, there is a danger that respondents have different perceptions of morale (Hardy, 2010). Consequently, Hardy (2010) suggests an approach for predicting the levels of morale from measuring a set of factors that influence morale. These antecedent factors of morale are divided into three main categories.

*Affective antecedents* consist of factors related to moods, feelings, emotions, and attitudes. For instance, high morale is often associated with positive affective states (e.g., sense of achievement, recognition, or happiness), while low morale is associated with negative affective states (e.g., feelings of failure, criticism, or injustice) (Hardy, 2010). The second category, *future/goal antecedents*, consists of factors related to the perception of individuals about the difference between the future and today as well as their goals for the future. For instance, believing in a better future, sense of progress, or success are considered as being key factors in increasing morale while lack of clarity, confidence, or progress are considered to lower morale (Hardy, 2010). Finally, the *interpersonal antecedents* of morale pertain to the factors related to the relationships and communication between individuals. For instance, having good relations with others, teamwork, or helping each other contributes to higher levels of morale while a bad work atmosphere or perceiving bullying or being dragged down by others can reduce morale (Hardy, 2010).

### 2.4. The relationship between technical debt and morale

The morale of software practitioners can be affected by several different things, such as, for example, the maturity or stability of the adopted development process (Lavallée and Robillard, 2012) or by the quality of the overall software product architecture (Jaktman, 1998). However, this study will only focus on the impact that TD has on morale.

Tom et al. (2013) argue that TD has a very strong negative effect on developers' morale. Following this argument, we tried to identify previous studies that discuss the relationship between TD and morale. In so doing, we used *Google Scholar* to search for studies mentioning both *Technical Debt* and *morale.* The search, which was conducted in January 2020, returned 415 potentially relevant results. After reading these results and checking their list of references, we were able to find only twelve sources mentioning the potential influence of TD on developers' morale or emotions (see Table 1).

We found three articles and one blog post that have mentioned the influence of TD on developers' morale. Also, during our literature search, we found four articles that mention the influence of

**Table 1**
Previous studies are mentioning the relationship between TD and developers' morale.

| Ref | Argument | Source of Argument |
|---|---|---|
| Codabux and Williams, 2013 | TD will hurt you later. It is better to pay upfront. | An Interviewee's opinion |
| Fernández-Sánchez et al., 2015 | Uncontrolled technical debt has a negative impact on the developers' morale. | Literature, Tom et al., 2013 |
| Lim et al., 2012 | Engineers don't like technical debt because they want to create perfect software. | An Interviewee's opinion |
| McConnell, 2010 June 14 | Working off debt can be motivational and good for team morale. | Author's opinion |
| Peters, 2014 | TD reduces developers' motivation. | Author's opinion |
| Spinola et al., May 2013 | Working off debt can be motivational and good for team morale. | Literature, McConnell, 2010 June 14 |
| Tom et al., 2013 | TD ultimately has negative impacts on morale. | Author's conclusion |
| | Developers would find the effects of technical debt frustrating in the long term. | Author's interpretation of a blog-post and several interviewees' opinions |
| | A lot of the tasks to prevent or repay the accrual of TD aren't very fun. | Several interviewees' opinion |
| Yli-Huumo et al., 2014 | Taking TD doesn't feel good for developers. | An interviewee's opinion |
| Suryanarayana et al., 2015 | Apart from technical challenges, technical debt also impacts the morale and motivation of the developer team. | Author's conclusion |
| Evans Data Corp and research, 2018 | 76% of the developers reported that paying down TD hurts their morale. | Result from a survey in a online- report |
| Ozkaya, 2019 | Messy code will drive messy behavior and frustration, limit understandability, and, consequently, induce stress in its developers and invoke a vicious cycle of bugs, vulnerabilities, and technical debt that is hard to get past. This cycle will eventually drive developer morale down. | Author's conclusion |
| Aldaeej, 2019 | The study finds that the negative impacts of TD in startups would be on morale, productivity, and product quality. | Literature, Giardino et al., 2016 |

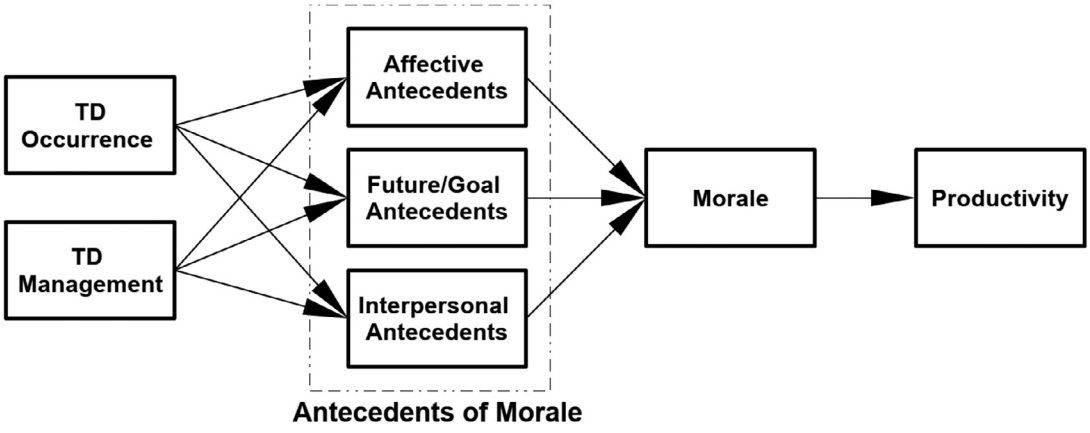



**Fig. 1.** The conceptual framework.

TD on developers' emotions. However, analyzing these sources revealed that only four of them support their arguments with empirical evidence, and the rest either cite other sources or report opinions. In addition, all these sources only mention the relationship between TD and developers' morale or emotions, and investigating this phenomenon is not their primary research focus.

Taken together, we aim to understand how the occurrence of TD and its management may relate the three dimensions of morale suggested by Hardy (2010). Therefore, we decided to conduct an exploratory field study to understand how TD and possibly its management could potentially influence software developers' morale. To achieve this, we had to adopt a research approach where we utilize the indirect approach suggested by Hardy (2010) to explore the potential influence of TD and its management on developers' morale (see Fig. 1). Using this approach not only enables us to investigate the impacts of TD and its management on developers' morale but also to gain a better understanding of the potential consequences of TD for developers and understand which dimensions of morale are more influenced by TD.

### *2.5. The relationship between technical debt and developer productivity*

Software developers' performance has a direct impact on software development productivity (Rasch and Tosi, 1992), and even if the term "waste of time" is commonly used within Lean organizations to target increased productivity, there are only a few studies looking into waste in software development organizations (Alahyari et al., 2019). As previously mentioned, several researchers describe that software suffering from TD has a negative impact on software development productivity. For instance,

Graziotin et al. (2018) state that affective states such as emotions, moods, and feelings have an impact on the productivity of individuals.

There is no commonly agreed definition of productivity (Sadowski and Zimmermann, 2019), but in software engineering, productivity is commonly defined, from a financial perspective, as the effectiveness of productive effort measured as the rate of output per unit of input (Sadowski and Zimmermann, 2019), (Scacchi, 1991), (Oliveira et al., 2017), (Ampatzoglou et al., 2015). Productivity is also a measure of the quality of an output relative to the input required to produce the output. This means that productivity is a combined measurement of efficiency and quality. However, Sadowski and Zimmermann (2019) argue that "*there is no metric that adequately captures the full space of developer productivity*" and instead, encourage the design of a set of metrics tailored for answering a specific goal and a purpose-based definition of the desired software value.

Several constraints can influence software development productivity, such as cost, schedule, and scope (Jensen, 2014). Moreover, Oliveira et al. (2017) express that researchers have not yet reached a consensus on how to measure productivity properly in software engineering. However, in this study, we have decided to focus on productivity in terms of the amount of wasted software development time due to developers being impeded during their daily software development work by experiencing TD within their software. However, we do acknowledge that lack of waste does not, by default, guarantee productivity. Our intention is not to redefine the term productivity to software practitioners and organizations, but we motivate this proxy based on the reason that if less time were spent on "extra" work tasks and activities due to experiencing TD, more time could be spent on other activities that would increase the overall productivity.

Sedano et al. (2017) echo this notion by stating that "w*aste is any activity that produces no value for the customer or user,*" and reducing this waste of time would thereby improve the software development productivity. In their study, they identified that TD could decrease the developer's productivity both in terms of wasted time and by causing reworking and an extraneous cognitive load.

This paper is somewhat related to a previous study we conducted (Besker et al., 2019). In that study, we more specifically studied the amount of wasted working time due to TD. The result of that study shows that TD contributes to the need to perform time-consuming additional activities and that developers waste, on average, 23% of all developers working time due to TD.

### 2.6. The relationship between morale and productivity

Previous research suggests that the level of employees' morale is correlated with their productivity (Damian and Chisan, 2006), (Fairley and Willshire, 2005), (Foulds and West, 2007), (Hardy, 2010), and project success (Hall et al., 2007). This correlation has been explained mainly in terms of employees' perception of their progress (Fairley and Willshire, 2005), (Hall et al., 2007), (Hardy, 2010), (Beecham et al., 2008), and their personal accomplishments (Damian and Chisan, 2006), (Foulds and West, 2007), (Maslach et al., 2001).

Thus, employees' progress is explained mainly in terms of their perception about the amount of work that is completed and the amount of unnecessary waste (Fairley and Willshire, 2005), (Hall et al., 2007), (Hardy, 2010). For instance, empirical data collected by Hardy (2010) suggests that individuals who perceive their morale too low consider the time spent on performing their work is less than the time they waste on organizing the workload. Fairley and Willshire (2005) suggest that software firms can significantly increase developers' morale and productivity by eliminating avoidable reworks (i.e., waste). A sufficient amount of rework (e.g., refactoring or testing) is known to be beneficial in software projects. For instance, Damian and Chisan (2006) suggest that developers' pride achieved as a result of continuous software improvements has a positive influence on developers' morale and ultimately leads to higher productivity. However, Fairley and Willshire (2005) argue that a majority of software firms deal with a significant amount of unnecessary reworks, which could have been avoided if the software development tasks had been performed correctly in the first place (e.g., by avoiding the unnecessary accumulation of TD). McConnell (2010) also suggests that such avoidable reworks are very costly and lead to a waste of development and maintenance time. Previous research suggests that TD taken reactively as a short-term solution such as minimizing documentation or testing or "dubious budget savings" (McConnell, 2010) lowers both morale and productivity (Foulds and West, 2007).

To conclude, several researchers have pointed towards the idea that working with software suffering from TD has an impact on the developers' morale and further that morale influences objective productivity indirectly. However, to the best of our knowledge, there is a lack of empirical studies assessing the relationships between TD, morale, and productivity.

The conceptual framework is shown in Fig. 1 and summarizes the above discussion. As illustrated in this figure, in this study, we aim to investigate the relationship between TD and developers' morale and its influence on their productivity. In particular, we try to understand how the occurrence and management of TD could affect the three dimensions of morale discussed above. Using this approach not only enables us to capture the attitudes of developers towards TD and its management but also to understand which dimensions of morale are more influenced by TD and its management. Additionally, we try to capture any potential correlational relationship between developers' morale and their productivity in terms of the amount of time they waste due to the occurrence of TD. To study the correlational relationships shown in our conceptual model, we have conducted an empirical study using a mixed-methods approach.

## 3. Methodology

Since our goal is to study TD and morale in its natural context, we have adopted a mixed-method approach where we use both qualitative and quantitative data collection and analysis methods, and part of the study also collects data longitudinally.

This study is based on data from 15 face-to-face interviews, and a survey, together with a longitudinal study, in order to examine the negative impact TD has on software developer morale and developer productivity. The study was conducted between September 2016 and January 2018.

As visualized in Fig 2, the overall research design was divided into six phases. The figure represents both activities that were performed in the initial study (darkest grayed boxes), the activities that were conducted partly in the initial study but enhanced in this extension of the study (light gray boxes), and also the additional research activities that are new in this part of the study (white boxes). The first four phases have a focus on answering RQ1 and RQ2, and when answering RQ3, all six phases are involved. Meaning that in phase 5 and 6, we collected data that we used together with the data from the previous phases when answering RQ3.

The following sections describe each phase and the related research methods used in each stage.

### 3.1. Phase 1—Contextual analysis and design

First, the study was presented and discussed during a workshop with software practitioners from several software companies, all

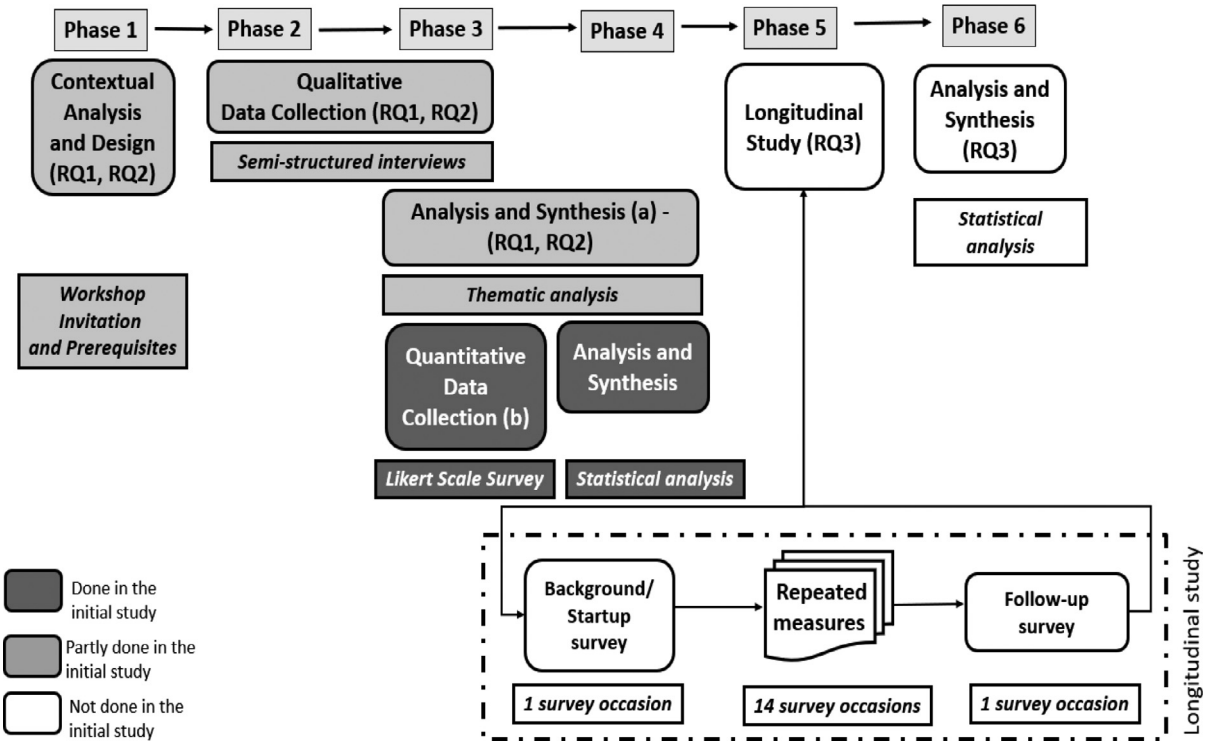



**Fig. 2.** Visualization of the research design and research method used in each phase.

having an extensive range of software development. The selection of companies was carried out with a convenience sample of industrial partners within our network. This phase acted as a guide for collecting data about the studied context and choosing the most suitable research design. The research team decided to base the research model on a longitudinal study together with supplementary follow-up interviews.

Secondly, an invitation to participate in the study was distributed to the workshop participants. Following the guidelines provided by Ployhart and Vandenberg (2009), to those 43 developers who approved to participate in the study, we emailed educational material (see Appendix D) intended to minimize inter-observer (all researchers communicate the same knowledge) and inter-instrument variability (all participants receive the same information).

Since the definition of TD is crucial and sets the context for the entire study, we specifically focused on the educational material on guiding the respondents to understand the basic concepts of TD fully. Since there is no unified description of TD, we selected three different descriptions of TD, in order to provide the participants with information that they could relate to from their own perspective. Even if the selected different descriptions provide similar information, they also illustrate different aspects of TD, such as consequences, artifacts, costs, and life-cycle perspectives. First, the initial and commonly used definition of TD was introduced by Cunningham (1992): *"Shipping first-time code is like going into debt. A little debt speeds development so long as it is paid back promptly with a rewrite… The danger occurs when the debt is not repaid. Every minute spent on not-quite-right code counts as interest on the debt,"* followed by Steve McConnell's also commonly used definition of TD (McConnell, 2010):*"A design or construction approach that's expedient in the short term but creates a technical context in which the same work will cost more to do later than it would cost to do now (including increased cost over time)."* Finally, a third description provided by the researchers of this study was used in order to provide a description of TD which illustrates that TD does not only have to include code- or design-related artifacts: *"Technical debt is a non-optimal solution in code (or other artifacts related to software development) that gives a short-term benefit but cause an extra long-term cost during the software life-cycle."*

### 3.2. Phase 2—Qualitative data collection

In the second phase, we conducted four rounds of interviews. This part of the study employed semi-structured interviews, including a mixture of open-ended and specific questions designed to elicit not only the information foreseen but also unexpected types of information (Seaman, 1999). The questions in the interviews were planned but not necessarily asked in the same order as they were listed. This interview technique allowed for the flexibility to explore interesting insights as they emerged. Each interview lasted between 30 and 45 min, and all interviews were digitally recorded and transcribed verbatim. All interviewees were asked for recording permission before starting, and they all agreed to be recorded and to be anonymously quoted for this paper.

We prepared a set of questions based on the antecedent factors of morale suggested by Hardy (2010), and the interview protocol is available in Appendix B. To improve the reliability of the collected data, at least two authors participated in the interviews. In these interviews, when possible, we asked the interviewees to show us at least one specific item from their TD backlog and answer our questions by considering that item. For example, we asked all the interviewees from two of the companies to pick some of the existing issues with its SonarQube code-analysis tool, while developers from another company were asked to choose some of the non-allowed dependencies highlighted by their in-house tool. When selecting these specific cases, we tried to verify that these TD issues were affecting the interviewees' work in practice.

### 3.3. Phase 3a—Analysis and synthesis

To analyze the qualitative data collected from interviews conducted in phase two, we used a thematic analysis approach

> "if you don't reduce the technical debt you will get development into a stall in some point; maybe we are actually walking now but we could run but the crawl will come if we don't pay-off the technical debt so we will get less and less feature in" – Interviewee 6

**Fig. 3.** Coded as "consequences of TD" and also as "Lack of progress," a code for low morale belonging to second-order theme "Progress" (see Appendix A).

(Braun and Clarke, 2006). Thematic analysis is a reliable data analysis method for capturing and reporting themes—important patterns of meaning related to a research phenomenon within qualitative data. Thematic analysis is especially suitable for studying the attitudes and behavior of people to explore a novel research phenomenon (Vaismoradi et al., 2013). In this study, we conducted a deductive thematic analysis, in which the researcher codes data according to a pre-existing coding frame rooted in previous theories (Braun and Clarke 2006). When analyzing the qualitative data, the guidelines provided by Braun and Clarke (2006) were used to conduct the analysis in three phases using a thorough and rigorous approach.

In the first phase, we prepared a codebook based on two main sources that were initially identified. First, using the theoretical framework suggested by Hardy (2010), a set of codes related to three dimensions of morale was generated.

Second, based on the results of Li et al.( 2015), a set of codes related to TD occurrence (i.e., Causes of TD and Consequences of TD) and its management was generated. For instance, the codes TD Identification, TD communication, TD measurement, TD monitoring, TD prevention, TD prioritization, TD repayment, and TD representation-documentation, corresponding to TD management activities suggested by Li et al. (2015), were capture how the interviewees manage TD. The full list of codes, second-order themes, and themes are shown in Appendix A. After preparing the codebook, the first author transcribed the recorded interviews, and the research team reviewed the transcriptions to familiarize themselves with the data and to get an overall idea of the collected data. The interviews with their transcriptions were added to a data analysis tool called NVivo.

In the second phase, the second author coded the interviews to identify data segments relevant to the research questions. Several of these initial codes were randomly picked and analyzed independently by the other authors, to triangulate the interpretation of the data and to minimize bias as much as possible. The coding procedure was reviewed by all the authors, and any conflicts were discussed jointly until an agreement was reached.

We continued the data analysis process by assigning the coded extracts of data to all the relevant themes. Each extract of data was assigned to at least one theme and, in many cases, to multiple themes. For example, the code shown in Fig. 3 was assigned to two second-order themes, "Progress" and "Consequences of TD," which are shown in Appendix A. Thus, we could capture the relationship between themes to create our initial thematic map. Later in this phase, the research team reviewed the identified themes, their relations, and coded data assigned to each of the themes. Based on the feedback from reviewing the themes, the second author continued to refine the codes and themes and the thematic map. During this phase, we realized that the data from the interviews did not support some of the original themes shown in Appendix A, and therefore, the thematic map was updated accordingly.

Finally, we prepared a discussion of the results of our data analysis to explain the relations between the occurrence and management of TD and the developers' morale. At the end of this phase, we put forward a set of propositions about the influence of TD and its management on antecedents of morale.

### 3.4. Phase 3b—Quantitative data collection

To complement our data and to clarify further the initial results from the interviews, we decided to conduct an online survey. In doing so, we designed a web survey that was hosted online by SurveyMonkey.com. The first draft of the survey was tested by the second author and one project manager to evaluate the understanding and the ordering of the questions and the usage of common terms and expressions (Czaja and Blair, 2005). During this evaluation, we also monitored the time that was needed to answer the questionnaire. The survey is shown in Appendix C.

The survey was accessible between December 1st, 2016, and January 31st, 2017, and a reminder were sent out after two weeks to all the invited participants. The survey was anonymous, and participation in the survey was voluntary. In designing the survey, we tried to form neutral questions and ordered and formulated them in a way that one question did not influence the response to the next question. The survey invitation was mailed directly to 34 developers in companies within our networks, all located in Scandinavia, with an extensive range of software development projects.

The first part of the survey gathered descriptive statistics to summarize the backgrounds of the respondents and their software. The second part of the survey included questions based on our theoretical propositions. As it can be seen from Table 2, the participants were asked to rate a set of 5-point Likert Scale statements (very slightly or not at all, a little, moderately, quite a bit, extremely), to indicate their opinion about the impacts of TD and its management on antecedents of morale.

The statements aimed at capturing the opinions of survey respondents about our interpretations of the interview data. As such, in creating the statements, we tried to look at the codes which were assigned to our second-order themes and choose clear keywords with a minimum chance of misinterpretation. For example, as can be seen in Appendix A, *Criticism* is a code that is assigned to the second-order theme called *Support and Communication,* which belongs to the *Affective antecedents* theme. Therefore, in ST1, we asked the participants whether they have been *criticized* by others for taking TD.

### 3.5. Phase 4—Analysis and synthesis

In the fourth phase, the data collected from the previous phase (3b) were analyzed quantitatively, that is, by statistical analysis of the data collected from the survey answers. For descriptive purposes, data were summarized by mean, median, and standard deviation for continuous variables and numbers and percentages for categorical variables. We also used statistical methods such as Kendall's tau-b and Spearman's rank correlation coefficient to assess the strength and direction of the association between different variables.

### 3.6. Phase 5—Data collection—longitudinal study

The goal of this phase was to collect information on how much working time developers report as wastage due to experiencing TD.

A longitudinal study is a research method involving the collection of repeated observations of the same variables (in our

**Table 2**
Statements rated by survey respondents.

| SID | Statements | The dimension of morale (second-order theme) |
|---|---|---|
| ST1 | I have been *criticized* by others for taking TD. | Affective (Support and Communication) |
| ST2 | I *feel confident* when I make a decision, which leads to TD. | Future/Goal (Vision for future) |
| ST3 | I feel that the presence of TD hinders me from *making progress*. | Future/Goal (Progress) |
| ST4 | I *feel upset* when others find out that I have taken TD. | Affective (Self-worth) |
| ST5 | I feel that *others appreciate* it when I pay back some TD. | Affective (Support and Communication) |
| ST6 | I *feel satisfied* when I pay back some TD. | Future/Goal (Progress) |
| ST7 | I am *encouraged by others* to pay back TD. | Interpersonal (Influence of others) |
| ST8 | I feel that paying back TD increases our team's morale. | All three |

case, the amount of wasted time) on more than one occasion (Ployhart and Vandenberg, 2009) and over time (Wohlin et al., 2000). The motivation for using this research method was mainly twofold:

a) We aimed to increase the precision of reporting experienced data (in our case, not based on single estimations or single perceptions of the wasted time). This was achieved by studying each respondent over several weeks using a repeated reporting design (Wohlin et al., 2000).
b) We aimed to survey respondents' changing reports over time: Longitudinal design has value for describing both temporal changes and their dependence on individual characteristics (Wohlin et al., 2000). Further, the longitudinal research method increases the precision of measuring and reduces inter-individual variation.

This data collection phase included three steps (with three individual and unique sets of surveys), where the first step focused on respondents' background data, the second on the amount of time the respondents wasted due to experiencing TD, and the third on developer morale (due to TD). SurveyMonkey.com hosted all quantitative data collection online during the longitudinal study.

**The first step** was a start-up survey collecting descriptive statistics to summarize the characteristics of the respondents and their companies.

**The second step** in the longitudinal phase collected repeated reporting of the wasted time due to experiencing TD. This stage was designed to collect reported data from 43 software developers at 14 different survey occasions (i.e., twice a week for seven weeks) from October to November 2016. In total, 473 data points were collected, and each respondent reported their wasted time, on average, 11 out of 14 occasions. In this step, the respondents reported their data (wastage of time) to an online survey twice a week. To have equal spacing between the reporting occasions, as suggested by Morrison (Morrison, 1970), for those respondents who did not answer within one day, a reminder was emailed.

During the entire period of this phase in the longitudinal study, the participants were asked to report their answer to the same survey question "*How much of the overall development time have you wasted due to technical debt (TD) since the last time you took the survey?*" Meaning the participant kept track of and calculated their own individual amount of wasted time.

In the surveys, the respondents reported the amount of wasted time using a value between 0–100% of their overall working time since they last took the survey. To address the potential problem with missing data from the respondents, if, for some reason, the respondents did not enter the data in one or more surveys, the respondents were asked to report their waste of time *since the last time* they took the survey. This means that if the respondent did not answer one or more surveys, the respondent would report the data from the last time the survey was taken. This means that reporting in the surveys cover the full period of sampling.

The **third step** of the longitudinal data collection phase was a follow-up survey to collect information indicating their opinion about the impacts of TD and its management on the antecedents of morale. This survey included questions based on our theoretical proposition presented in Section 4. In this part of the survey, participants were asked to rate a set of 5-point Likert Scale statements (very slightly or not at all, a little, moderately, quite a bit, extremely).

### 3.7. Phase 6—Analysis and synthesis

The data collected in the sixth phase were analyzed quantitatively, that is, by interpreting the numbers collected from the survey answers. All statistical analyses were performed with SPSS (version 22) and R version 3.3.2, using Tidyverse (Wickham, 2009) version 1.1.1.

During this phase, we collected 473 data points (e.g., reporting of the amount of wasted time), where each developer reported on average 10.73 times out of 14 possible occasions (with a median of 12 and std. dev. of 3.91 times) with the average time interval between the reporting occasions of 3.1 days (with a median of 2.7 and std. dev. of 1.3 days).

In this phase, we further analyzed the association between the wasted working time and the antecedents of morale (Section 2.2). The goal of this statistical analysis was to primarily indicate correlation, not causality, where the purpose was to understand if the quantity of wasted time, directly generated by TD, was associated with how the respondents perceived TD with respect to morale. For example, were respondents more inclined to consider TD affecting their morale if they wasted more time because of it? This was a way to analyze additional, quantitative, evidence to support (or not) our results. Such a step could be considered as increasing source triangulation of our study, improving its validity. Accordingly, for the test of normality (see Section 4), we used a non-parametric method, called Spearman coefficient rank.

## 4. Results

The following subsections present the results for the research questions presented in Section 1. First, we present the results from the analysis of the qualitative data collection for each of the three identified dimensions of morale. Secondly, we present the results from the quantitative data collection.

### 4.1. Qualitative data—Influence of TD on morale (RQ1, and RQ2)

In total, we conducted 15 face-to-face interviews with software professionals from five different companies (see Table 3). Of the 15 interviewed respondents, we gained access to information about their working experience within the software development field. Most of the interviewees were relatively experienced, with a minimum of 5 working years, a maximum of 44 years and a median of 11 years.

Company A develops equipment for aerospace crews, and company B is an international telecom company. Both companies are

**Table 3**
Characterization of Respondents.

| ID | Role | Experience (years) | Company | Country | Domain |
|---|---|---|---|---|---|
| I1 | Software Developer | NA* | Company A | Sweden | Aerospace |
| I2 | Software Developer | 28 | | | |
| I3 | Software Developer | 44 | | | |
| I4 | Software Developer | 7 | Company B | Sweden | Telecom |
| I5 | Software Developer | 8 | | | |
| I6 | Software Developer | 26 | | | |
| I7 | Testing Engineer | 11 | Company C | Finland | Healthcare |
| I8 | Software Engineer | 6 | Company D | German | Aerospace |
| I9 | Software Developer | 21 | Company E | Portugal | Healthcare |
| I10 | Lead developer | 10 | Company E | Portugal | General IT |
| I11 | Software Developer | 9 | Company E | Portugal | Telecom |
| I12 | Project Manager | 13 | Company E | Portugal | Gaming |
| I13 | Senior Architect | NA* | Company E | Portugal | Telecom |
| I14 | Software Developer | 11 | Company E | Portugal | General IT |
| I15 | Software Developer | 5 | Company E | Portugal | Gaming |

* The interviewee preferred not to share the work experience.

located in Sweden. Company C is a medium-sized Finnish firm creating and providing a wide range of clinical data-assessment solutions to customers active in the healthcare sector. Company D is a German branch of an international enterprise providing software development and maintenance services to major aerospace customers such as the European Space Agency. Finally, company E is a Portuguese SME company providing a wide range of IT solutions to both public and private sectors in different areas such as telecommunications, gaming, finance, and so forth. The size of the companies cannot be presented due to the risk of violating their anonymity.

The results from the interviews suggest that both the presence and management of TD influence developers' morale to some extent. In particular, the occurrence of TD mainly has a negative influence on the affective and future/goal antecedents of morale. However, our results do not show any considerable impact of TD occurrence on the *Affective* and *Interpersonal* dimensions of morale. During the thematic analysis, we realized that the interview data do not provide enough evidence to saturate some of the original second-order themes and main themes related to TD occurrence. These themes and their relationships are shown in gray in our final thematic map in Fig. 4.

On the other hand, the interview data indicate that the management of TD has a strong positive influence on morale. In particular, the interview data show that TD management has a clear positive influence on all the second-order themes and consequently forming the three dimensions of morale. In the following sections, we discuss these findings in more detail.

#### 4.1.1. Affective antecedents (RQ1 and RQ2)

To explore the influence of TD and its management on the affective dimension of morale, we assigned codes into three second-order themes (i.e., valued and taken seriously, self-worth, and support and communication). However, from the interviewees' data, we could only identify a clear relationship between the occurrence of TD and self-worth and a weak link to support and communication. In other words, it seems that the occurrence of TD only has a negative influence on developers' self-worth, but they do not have any strong affective reaction to it.

Most of the interviewees mentioned that they do not think that their colleagues or management team have ever criticized them for introducing TD. However, some of them mentioned that since being criticized could be unpleasant, an individual developer might hesitate to criticize their teammates for taking on TD. Thus, it is very important to communicate others' mistakes appropriately. Interviewee 3 raised this issue as shown in the following excerpt: "*You would have to go and ask them why it is all red in your [SonarQube] panels when mine is green, but that is a question that is a bit difficult to ask if people get upset if you hint that they have bad quality.*"

Even though most of the interviewees think that they have not been subject to criticism and they should not blame others for taking on TD, several interviewees mentioned that they do not feel good about making a decision, which obviously leads to the occurrence of TD. The experience of realizing that they have introduced TD is explained by several interviewees as a negative feeling by using a wide range of negative terms such as a sense of insecurity and failure or being afraid and upset. For instance, interviewee 2 mentioned this issue this way: "*It is some kind of a bad feeling because it nags me. […] if I have missed something that I feel that I should have known.*" Several interviewees mentioned that the reason for such feelings is that they tend to perform a good job. On the other hand, some of the interviewees mentioned that working with software artifacts, especially legacy code, which contains a large amount of TD is scary and frustrating. It must be noted that three interviewees mentioned that such negative feelings might depend on the level of work experience. Based on the above discussion, we put forward the following proposition[1]:

***PTD1: The occurrence of TD may have a negative influence on the affective antecedents of morale.***

On the other hand, it seems that the management of TD influences all the second-order themes of affective dimension. First, the majority of the interviewees mentioned that their efforts for managing TD are valued and appreciated by their colleagues and management team. For example, interviewee 4 mentioned that "*if you deal with technical debt that gives a measurable surface impact for improvement […] those things are highly appreciated*".

Additionally, the majority of our interviewees think that others trust their decisions concerning TD management, while half of them mentioned that they have a reasonable amount of autonomy to perform refactoring and repaying TD. However, it seems from the interviewees' perspective that the companies do not reward TD management sufficiently. One developer even mentioned that in their company, slight praise or appreciation is missing, while another said that if developers cannot show its value and results, others do not appreciate repaying TD.

On the other hand, the interview data indicate that managing TD is associated with a sense of self-worth among software developers. The majority of the interviewees think that the identification and communication of TD enable them to improve their skills by understanding their mistakes and learning new aspects of

[1] In acronyms used for propositions, PTD stands for Proposition Technical Debt and PTDM stands for Proposition Technical Debt Management.

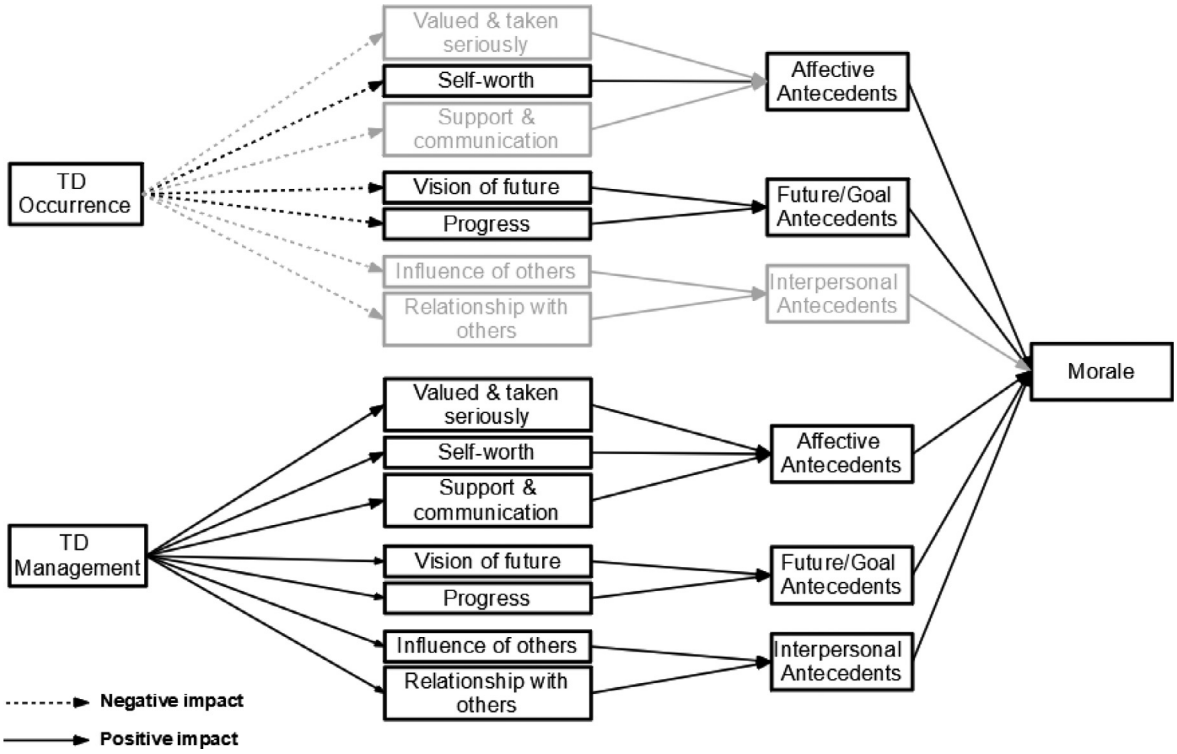



**Fig. 4.** Visualization of the final thematic map.

software engineering. Interviewee 1, for instance, mentioned that "*over the years [it had] been teaching me lots of stuff and made me a better developer in a shorter time than would have been possible without it. I think it has been a huge change*." Also, it seems that managing TD leads to a sense of achievement and success among developers. Several interviewees mentioned that performing refactoring and repaying TD is an interesting activity, which motivates them as well as enables them to increase the quality of software artifacts constantly and feel confident about their products.

However, several interviewees mentioned that managing TD is not an easy and straightforward task, and it may be considered pointless. Managing TD of legacy code is especially challenging since developers often need to perform a long chain of unexpected changes in different parts of the code until they get to fix the initial issue. Also, it is hard to show the real value of such improvements and convince managers why it is necessary to perform them. Several interviewees considered repaying TD to be unnecessary or even dangerous in some cases since it might cause failure in a working system. Therefore, some developers might prefer not to touch the code since they consider repaying TD to be pointless.

Finally, the majority of our interviewees mentioned the good support and communication within their company for TD management. It seems that they consider TD as an inevitable phenomenon, which must be identified, documented, and repaid as time and resources allow. In other words, no matter who introduces TD, they consider TD management a task that everyone can benefit from and necessary for improving the quality of the software artifacts.

Overall, based on the observations discussed in this section, we propose that TD has a negative influence on affective antecedents of morale, while TD management has a positive influence on affective antecedents of morale. Overall, based on these observations, we propose that:

***PTDM1: TD management has a positive influence on the affective antecedents of morale.***

*4.1.2. Future/Goal antecedents (RQ1 and RQ2)*

To explore the influence of TD and its management on the future/goal dimension of morale, we assigned codes into two second-order themes (i.e., the vision of future and progress). Based on the interview data, the influence of TD on the future/goal antecedents of morale seems to be strong. The majority of the interviewees mentioned that the presence of TD has a negative influence on their progress and future activities. In particular, they mentioned that the presence of TD hinders them from progressing in their tasks. At the same time, it makes the future unclear since developers may face unpredicted difficulties.

As mentioned by several interviewees, large amounts of TD make the maintenance difficult and costly since understanding and working with the software becomes problematic, and there might be unpredictable issues. Also, if TD is not paid back, the development speed can decrease, and adding new features in the future becomes very challenging and even impossible. In the following excerpt, interviewee 6 talks about such an issue*: "If you don't reduce the technical debt, you will get development into a stall in some point; maybe we are walking now, but we could run, but the crawl will come if we don't pay off the technical debt."*

On the other hand, some of the interviewees mentioned that the presence of TD brings developers' confidence down. This could either be because of their awareness of a large volume of TD in their artifacts or because it hinders developers from performing their tasks. Several interviewees mentioned that introducing TD could make them feel that they lack the necessary skills and lower their confidence. As a result, they start to be more conscious about their decisions, and therefore, development speed slows, or developers will be reluctant to perform refactoring to improve their code.

Several interviewees mentioned the influence of past decisions leading to the occurrence of TD on their progress. Even though they did not blame previous developers for taking on TD, some of the interviewees mentioned that they have to deal with TD, which

is the result of poor decisions made by previous developers and which could have been avoided. For instance, interviewee 8 mentioned that "*in the past, either we or that company before us did not really pay attention and violated the guideline, and because of that, we have to pay back [TD] now.*"

Also, several interviewees mentioned that when different modules are developed by different developers or teams, the existence of TD in one module can influence the progress of other developers as well. As a result, working with an external module, which is not well maintained, is often frustrating and time-consuming. Based on the above discussion, we put forward the following proposition:

***PTD2: The occurrence of TD has a negative influence on the future/goal antecedents of morale.***

On the other hand, the majority of the interviewees mentioned the importance of TD management, which, from their perspective, makes the future better than the present. These interviewees mentioned that they have a clear vision of how to manage TD properly within their company. In doing so, they use different techniques and tools to identify, document, communicate, prevent, and repay TD. Also, it seems that proper TD management is associated with a sense of satisfaction. For instance, interviewee 6 mentioned that *"I think that is a nice feeling, to be able to pay your debt […] I would say it is a positive feeling; it is a motivator."*

All the interviewees consider repaying TD to be necessary, but for some items, it is pointless. For instance, several interviewees mentioned that repaying architectural TD and testing TD is very important, while some of them mentioned that it is pointless to fix issues such as documentation TD, which they considered to be "cosmetic" or the items which are mistakenly highlighted by tools (i.e., false positives), or TD items located in those parts of the legacy code that rarely change. However, during the data analysis process, we identified controversial opinions about such "cosmetic" things and "false positives." For instance, one software developer suggested that following simple rules such as naming conventions is very important, while another one suggested that it does not make sense to go back and fix naming issues. In another case, one interviewee suggested that writing good comments facilitates future maintenance of software, while another one mentioned that repaying documentation TD is not worth spending the time and effort.

From the interview data, it seems that proper TD management leads to a sense of progress among software developers. The majority of the interviewees consider TD identification and communication a contribution to their teams' goals, and by repaying TD, they feel successful in progressing toward those goals. Such a sense of progress is a positive feeling, which enables developers to perform their tasks easily and smoothly.

Therefore, based on the above-mentioned observations, we propose that the occurrence of TD has a negative influence on the future/goal antecedents of morale, while TD management has a positive influence on the future/goal antecedents of morale. Therefore, based on these observations, we propose that:

***PTDM2: TD management has a positive influence on the future/goal antecedents of morale.***

*4.1.3. Interpersonal antecedents (RQ1 and RQ2)*

To explore the influence of TD and its management on the interpersonal dimension of morale, we assigned codes into two second-order themes (i.e., the influence of others and relationships with others).

Regarding the relationship between the influence of others and the occurrence of TD, the majority of our interviewees do not blame their colleagues or previous developers for introducing TD even though they acknowledge that previous sub-optimal decisions which have led to the occurrence of TD could be avoided in the past. The following excerpt shows the opinion of interviewee 5 in this regard: *"It maybe is stressful or frustrating to work with this legacy code; if [it] had been written in a better way it would have been much faster and easier to implement this feature."*

Regarding the relationship with others, it seems that from the interviewees' perspective, there is very good cohesion and understanding within their teams about the reasons behind the occurrence of TD. Most of the interviewees think that a combination of factors leads to the occurrence of TD, and therefore, they do not feel that their teammates or previous developers dragged them down by introducing TD. This seems to be the case, especially in those teams where team members have a closer relationship, as interviewee 10 mentions in the following excerpt: *"We should not point a finger or place a picture on the wall, 'this guy broke some rules.' And I also believe that a significant part of company's success is directly motivated by [and] directly related to our engagement as colleagues in our team because we say we are all friends here; we know each other; three of us know each other since college days, and we are friends also outside."*

Our interview data suggest that the decision to take on TD is often made based on a consensus among team members, especially in smaller teams. For instance, one interviewee stated that "*if we have a short time frame sometimes [and need to] make a decision, do we go to technical debt and let these things go like this so we can keep the deadline or not? We discuss that usually. In the end, if the team has a consensus, there are no problems, but if there is no consensus in the team, the lead developer—the senior one—makes a deciding vote."*

Since we could not discover strong evidence from our data indicating that TD has a negative influence on interpersonal antecedents of morale, we put forward the following proposition:

***PTD3: The occurrence of TD has no negative influence on the interpersonal antecedents of morale.***

On the other hand, regarding the influence of TD management on the interpersonal dimension of morale, it seems that the majority of our interviewees consider TD management as teamwork by which they contribute to a set of common goals within their teams. They especially consider conducting reviews as a positive contribution by which they can identify TD and help each other in improving their skills and, ultimately, the quality of software. In the following excerpt, interviewee 10 refers to this matter: "*Well, I appreciate that review. I always try to learn, and learning from my mistakes is also something that I do when I keep doing some self-evaluation of my team, and that's one way my evaluation refines. Yes, I appreciate that someone evaluates my work generally, and the code is just one of the parts.*"

On the other hand, regarding the relationship with others, most of our interviewees think that the current team is responsible for managing TD and improving the quality of software constantly, even though a few of them mentioned that it is not pleasant to improve the legacy code. This may be due to the challenges of repaying the TD of the legacy code, as discussed earlier. Therefore, they think that identification and communication of TD is a service, which is done to help their teammates in improving the quality of software artifacts. In the following excerpt, interviewee 2 points to this: "*I will [report it] so they have to fix it or to have a more thorough discussion about it, why it happened and why you did that if it can't be fixed. But mostly I feel that it is a service; we do each other that kind of service that together we can make better code*". Based on the above discussion, we propose that:

***PTDM3: TD management has a positive influence on the interpersonal antecedents of morale.***

To summarize, the interview data provide evidence that TD can have a negative influence on affective and future/goal antecedents of morale but no influence on interpersonal antecedents of morale. On the other hand, TD management has a positive influence on all

**Table 4**
Characteristics of the sample survey.

| Individual | Company |
|---|---|
| **Experience** | **Software system type(s)** |
| < 2 years 8.8% | Embedded Sys. 76.5% |
| 2 - 5 year 17.6% | Real-time Sys. 29.4% |
| 5 - 10 year 17.6% | Data management Sys. 5.9% |
| > 10 years 55.9% | System Integration 2.9% |
| **Education** | Modeling and/or simul. 2.9% |
| Master's degree 76.5% | Data analysis sys. 14.7% |
| Bachelor's degree 20.6% | |
| Ph.D. degree 2.9% | |
| **Gender** | |
| Male 82.3% | |
| Female 17.7% | |

three dimensions of morale, especially on the future/goal dimension.

### 4.2. Quantitative data—Influence of TD on morale (RQ1, and RQ2)

In total, we received 34 valid responses to the survey from developers across seven software companies. Please note that two of these respondents work at Company A, five work at Company B, and the rest (i.e., 27 respondents) work in other companies from which we did not collect any interview data.

Initially, the survey gathered descriptive statistics to summarize the backgrounds of the respondents and their software. This data is compiled and presented in Table 4. As can be seen from this table, all the survey respondents have completed a university degree, while the majority of them have more than 10 years of work experience in the software industry.

As illustrated in Table 4, all the respondents were relatively experienced as software developers: 56% had more than 10 years of experience, and only 7% had fewer than 2 years of experience. All respondents have a university-level education, where 82% have a master's degree.

Further, the results from the survey assessing the impact of TD on the different dimensions of morale show that the occurrence of TD negatively influences the future/goal and affective antecedents of morale but not its interpersonal antecedents, while the management of TD positively affects all the three dimensions of morale.

When studying the correlation between the amount of time that is wasted due to TD and the developer morale, our result shows that the more time that is wasted, the more the respondents feel that the progress is hindered by TD. In the following sections, we discuss these findings in more detail.

#### 4.2.1. Survey result (RQ1 and RQ2)

As described in Section 3.1, we complemented our findings from the interviews by conducting a survey in which 33 software professionals rated a set of 5-point Likert Scale statements (see Table 5). Utilizing our theoretical framework and findings from the interviews, we prepared these statements in a way that they complement our findings from the interviews. As illustrated in Table 5, four of these statements are related to the negative effects of TD (ST1, ST2, ST3, ST4), and three of them are related to the positive effects of TD management (ST5, ST6, ST7) on antecedents of morale. Finally, ST8 directly refers to the positive influence of TD management on morale. This enables us to verify whether respondents' ratings about the influence of TD management on antecedents of morale are in line with their perception of the influence of TD management on morale itself. Fig. 5 shows a summary of the ratings provided by the respondents.

Table 5 shows the central tendency (i.e., median) and frequency (i.e., interquartile range) of the respondents' ratings. It must be noted that a greater median indicates that the respondents consider the influence mentioned in a statement to be more significant, while a smaller median indicates such an influence to be less significant. On the other hand, a lower interquartile range (IQR) value indicates higher levels of consensus among respondents, while a higher IQR value implies lower consensus among respondents.

Regarding the influence of TD on the affective antecedents of morale, ST1 received a very low rating (median = 1) with a very high level of consensus among respondents (IQR = 0). This suggests that the respondents think that they rarely have been subject to criticism for introducing TD. Moreover, ST4 received a very low rating with a high level of consensus among respondents (median = 1; IQR = 1). These observations suggest that the respondents consider the affective consequences of TD to be very insignificant, which is in line with our interview data. Therefore, we can conclude that the occurrence of TD has a negative influence on developers' morale since it reduces their confidence.

Regarding the influence of TD on the Future/Goal dimension of morale, ST2 received a moderate rating (median = 3), which suggests that the majority of respondents, with a medium level of consensus (IQR = 2), do not feel very confident when introducing TD. Moreover, the ratings for ST3 (median = 3; IQR = 2) show that the survey respondents, with a medium level of consensus, consider TD to hinder them moderately from making progress. These results support our findings from interviews, which indicate that the occurrence of TD has a negative influence on Future/Goal antecedents of morale. In other words, we can say that the occurrence of TD primarily has a negative influence on developers' morale because it hinders their progress.

On the other hand, concerning the impacts of TD management on antecedents of morale, ST5 received a high rating (median = 4) with a medium level of consensus among respondents (IQR = 2). This observation supports our interview data and suggests that the survey respondents generally think that their efforts for managing TD are appreciated by their colleagues. Therefore, it can be said that TD management has a positive influence on developers' morale since it is associated with a sense of appreciation. In addition, ST6 received a high rating from respondents with a high level of consensus (median = 4; IQR = 1), which indicates that the respondents strongly feel satisfied with repaying TD. This supports our interview data, which suggests managing TD has a positive influence on future/goal antecedents of morale since it is associated with a sense of progress. In other words, TD management has a positive influence on developers' morale since it is associated with a sense of progress.

As can be seen from Table 5, ST7 received a moderate rating (median = 3) with a medium level of consensus (IQR = 2). This shows that the respondents think that they are moderately encouraged by others to repay TD, which indicates the positive influence of TD management on interpersonal antecedents of morale. This observation is in line with our findings from the interviews. In particular, we can conclude that TD management has a positive influence on developers' morale since it is encouraged by others. Finally, ST8 was commonly rated as significant (median = 4; IQR = 1), which alongside interview data, suggests that, in general, management of TD has a positive influence on developers' morale. Based on these results, we are confident that our propositions about the influence of TD management on different dimensions of morale are plausible, and therefore we put forward another proposition as:

***PTDM4: TD management is perceived to have a positive influence on the team's morale.***

To summarize, the occurrence of TD negatively influences developers' confidence and their sense of progress and consequently lowers their morale. On the other hand, management of TD is as-

**Table 5**
Median and IQR calculated for survey responses.

| SID | Statement | Median | IQR | Supports the interviews |
|---|---|---|---|---|
| ST1 | I have been criticized by others for taking TD. | 1 | 0 | Yes |
| ST2 | I feel confident when I make a decision, which leads to TD. | 3 | 2 | Yes |
| ST3 | I feel that the presence of TD hinders me from making progress. | 3 | 2 | Yes |
| ST4 | I feel upset when others find out that I have taken TD. | 1 | 1 | Yes |
| ST5 | I feel that others appreciate it when I pay back some TD. | 4 | 2 | Yes |
| ST6 | I feel satisfied when I pay back some TD. | 4 | 1 | Yes |
| ST7 | I am encouraged by others to pay back TD. | 3 | 2 | Yes |
| ST8 | I feel that paying back TD increases our team's morale. | 4 | 1 | Yes |

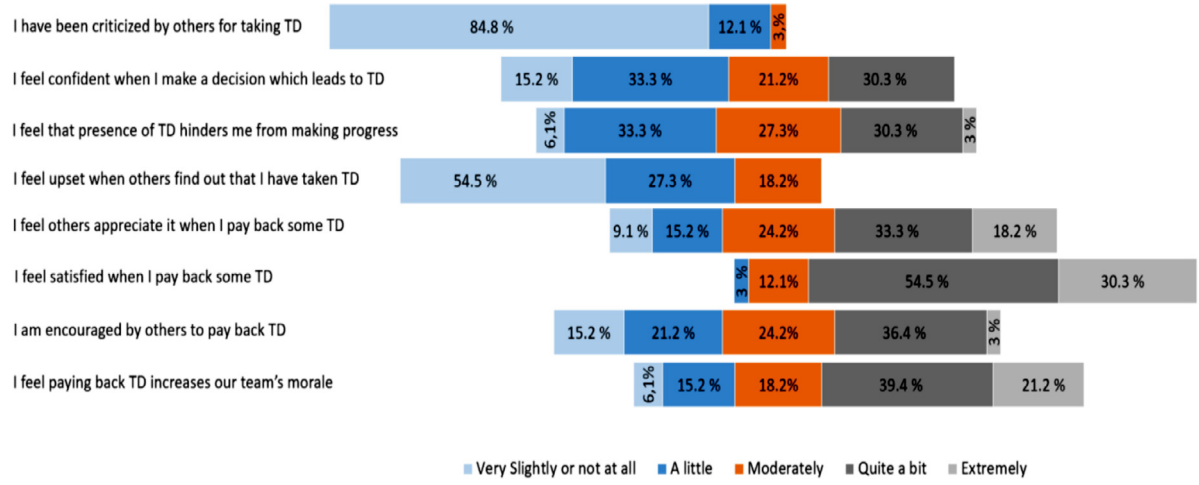



**Fig. 5.** Summary of the responses to the online survey.

**Table 6**
Test of Normality for Correlated Variables.

| Variable | Statements | Shapiro-Wilk test | p-value |
|---|---|---|---|
| Waste | Wasted time because of Technical Debt | 0.80817 | 3.701e-05 |
| ST1 | I have been criticized by others for taking TD. | 0.86125 | 0.0005045 |
| ST2 | I feel confident when I make a decision, which leads to TD. | 0.74087 | 2.221e-06 |
| ST3 | I feel that the presence of TD hinders me from making progress. | 0.89586 | 0.003573 |
| ST4 | I feel upset when others find out that I have taken TD. | 0.48118 | 7.893e-10 |
| ST5 | I feel that others appreciate it when I pay back some TD. | 0.88216 | 0.001602 |
| ST6 | I feel satisfied when I pay back some TD. | 0.82243 | 7.176e-05 |
| ST7 | I am encouraged by others to pay back TD. | 0.90563 | 0.00648 |
| ST8 | I feel that paying back TD increases our team's morale. | 0.88989 | 0.002506 |

sociated with a sense of progress and a sense of appreciation and support from others, which raises developers' morale. To further investigate these findings, we performed a set of statistical tests to identify any potential correlations between our propositions. This would show if and what kind of relationships there might be among the different dimensions of morale affected by TD occurrence or management as well as to verify any potential influence on our results of two additional factors (i.e., developers' level of work experience and the amount of waste that occurs due to TD) identified from the interview data.

Before analyzing the association between wasted time and the antecedents of morale, we first tested the variables for normality using the Shapiro-Wilk test of normality (see Table 6). None of the variables are normally distributed, as the null hypothesis (i.e., the data points of the variable are extracted from a normally distributed population) is rejected with very low p-values. This means that we cannot use Pearson, but we need to use non-parametric methods. Therefore, we use the Spearman's rank correlation coefficient (see Table 7).

The matrix of correlation in Fig. 6 shows only those associations that are significant (p-value $< 0.1$). The blank cells represent no correlations.

We can see how most of the statements related to the occurrence of TD and morale were not statistically correlated among themselves or with the statements related to TD management. On the other hand, we can see how most of the statements related to the management of TD and morale are correlated among themselves. These results are further discussed in the following paragraphs.

- *The negative correlation between criticism (lack of support and communication) and sense of progress (ST1-ST6):* This significant negative correlation ($-0.29$, p-value $< 0.1$) indicates that the less criticism the practitioners receive because of taking TD, the more satisfaction they get from their progress in paying it back. On the contrary, this negative correlation may also be interpreted as though developers feel less satisfaction in paying TD back, but they might also perceive more criticism due to taking TD.
- *Correlation between a sense of appreciation and sense of support (ST5 - ST7):* This strong and significant (0.77, p-value $< 0.05$) correlation indicates that practitioners who feel encouraged to repay TD also feel that their efforts towards TD repayment are recognized by their colleagues. This could potentially mean that

**Table 7**
Correlation between the waste of time and the statements.

| % Waste correlation with: | ST1 | ST2 | ST3 | ST4 | ST5 | ST6 | ST7 | ST8 |
|---|---|---|---|---|---|---|---|---|
| Spearman (approx.) | −0.109 | −0.036 | **0.453** | 0.061 | −0.076 | 0.087 | −0.202 | −0.070 |
| p-value (approx.) | 0.536 | 0.837 | **0.007** | 0.728 | 0.666 | 0.624 | 0.251 | 0.690 |

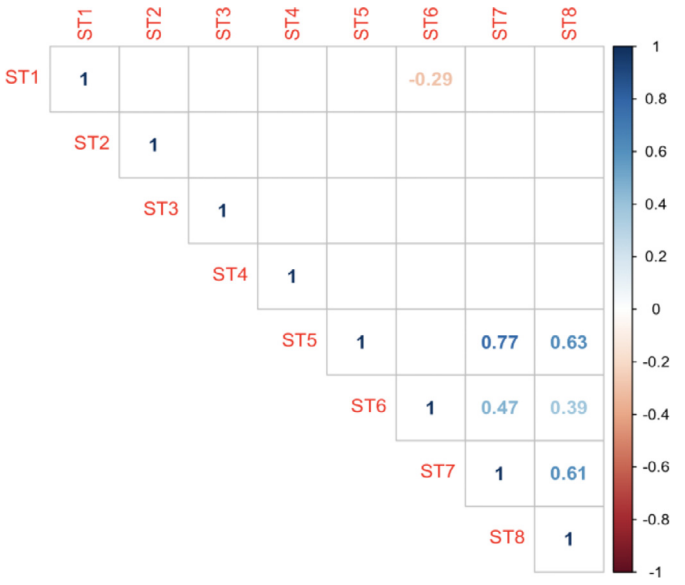



**Fig. 6.** Associations between the different statements.

showing appreciation and recognition is a good method for encouraging developers to repay TD.

- *Correlation between a sense of progress and sense of support (ST6-ST7):* This medium correlation (0.45, p-value $< 0.05$) can be interpreted as the more encouraging support the respondents receive to repay TD, the more satisfied they are with their progress. The inverse implication could potentially be that if someone feels satisfied by paying TD back, they also feel encouraged by others to do so. On the other hand, since satisfaction could also be considered as an affective state which is associated with self-worth (i.e., second-order theme in affective antecedents), this correlation might be interpreted as the more encouraging support the respondents receive to repay TD, the better they feel about their achievements.
- *Correlation of sense of appreciation, sense of progress, and sense of support with the perceived team's morale (ST5, ST6, ST7, and ST8):* We discuss these correlations altogether because ST5, ST6, and ST7 are already correlated and have been discussed. From these correlations, we could infer that those practitioners who were encouraged to pay TD back, who were satisfied in doing so (although the correlation is less strong here) and felt like others would appreciate it, also think that paying back would increase team morale. One explanation might be that potentially all these factors contribute to better team morale or having better team morale makes others feel like they are encouraged, satisfied, and supported in repaying TD. In the second case, it could potentially be explained as "*we are happy in the team, so we feel like we are better at dealing with TD*," which might also be a possible explanation, although less supported by evidence from interviews.

As discussed earlier, some of the interviewees mentioned that developers' perception of the consequences of TD could depend on their level of work experience. To determine such relationships, a set of *Kendall's tau-b* correlation tests were conducted. The results of these tests indicate that the only considerable correlation is between respondents' work experience and their perception about being criticized for introducing TD (i.e., ST1), which was positive and statistically significant at the 0.05 level ($\tau$b $= 0.338$, $p = 0.039$). This rejects the idea that less-experienced developers are more concerned about being subject to criticism for introducing TD and supports our previous findings indicating that, in general, developers rarely criticize each other for the occurrence of TD. Another interpretation could potentially be that the more experienced developers perceive a higher level of criticism in case others highlight their mistakes or their actions in taking TD.

*4.2.2. Correlation between the wasted time (due to TD) with the TD occurrence and the TD management dimensions of morale (RQ3)*

As discussed earlier, almost all our interviewees and respondents to the 5-point Likert Scale survey mentioned that the occurrence of TD hinders their progress and their future activities. Several interviews discussed the amount of development time and resources that are wasted because of the TD accumulated in their products. Therefore, we decided to investigate any potential correlation between the amount of waste that occurs due to TD and developers' morale.

First, we calculated the average amount of perceived wasted time reported by each participant during the second step of the longitudinal data collection (i.e., phase 5). Following this, we used Spearman ranking to study the correlation between the average amount of wasted time reported by each respondent, and their rating for the 5-point Likert scale statements. The association might tell us whether participants who reported a higher average waste of time would also rate higher the statements ST1-ST8 (the antecedents of morale related to the occurrence and management of TD). In other words, we want to understand whether more waste (the negative effect of TD) correlates with lower or higher morale (with respect to different dimensions)

As we can see from Table 7, the only significant moderate correlation is between the average% of the wasted time and ST3 ("I feel that the presence of TD hinders me from making progress"). This indicates that the more time is wasted, the more the respondents feel that the progress is hindered by TD. This interpretation is supported by our observations from the interviews. However, the observed correlation can also be interpreted differently: The respondents with a lower level of progress, or possibly morale, might tend to estimate the amount of waste to be higher.

The lack of associations between the waste due to TD and the morale variables may not exclude the existence of relationships. But in this study, we do not find evidence indicating that the waste of time due to TD correlates with other morale variables than ST3.

## 5. Discussion

In this study, we aim to answer and discuss the earlier stated research questions. In response to *RQ1*, regarding the influence of TD on morale, it can be said that the occurrence of TD may reduce developers' morale mainly because the developers perceive it to hinder their progress and makes it challenging to perform their tasks. In response to *RQ2*, it can be said that proper management of TD appears to increase developers' morale since it is associated with positive personal and interpersonal feedback and enables developers to perform their tasks better and to improve software

quality in the future. In *RQ3*, we assess how the amount of wasted time correlates with developer morale, and the results indicate a significant correlation between the productivity (e.g., the perceived wasted time) and the developers' feeling that their progress is hindered by TD.

Although in recent years, considerable interest has been shown in researching TD, there is a lack of studies exploring this phenomenon from individuals' perspective and particularly concentrating on explaining the social and psychological aspects of TD. By reviewing previous literature on TD, we could find only a limited number of previous studies mentioning that TD may influence developers' emotions (Yli-Huumo et al., 2014), (Codabux and Williams, 2013), and morale (Spinola et al., 2013), (Tom et al., 2013), (Fernández-Sánchez et al., 2015), (Peters, 2014). Therefore, we decided to conduct a field study to explore the potential impacts of TD and its management on developers' morale.

Our findings show that the occurrence of TD is perceived to have a negative influence, mainly on the future/goal and affective antecedents of morale and not on its interpersonal antecedents. In other words, our results indicate that TD reduces developers' morale since it hinders them from performing their development tasks, making progress, and achieving their goals. This is in line with previous studies by Codabux and Williams (2013) that suggest the presence of TD slows development and reduces developers' productivity (Tom et al., 2013). Also, the lack of development resources is reported to hinder developers' progress and seam to consequently lower their morale (Hall et al., 2007), (McConnell, 2010). These results further support our findings considering that resource constraints are one of the main reasons for incurring TD (Ghanbari et al., 2018).

Our results also indicate that the occurrence of TD has a negative influence on affective antecedents of morale and, in particular, developers' self-worth. In an exploratory field study and based on interviews with 35 software developers, Lim et al. (2012) suggest that software developers have a negative attitude toward TD since they tend to create "*perfect software.*" The results of another case study conducted by J. Yli-Huumo, Maglyas and Smolander (Yli-Huumo et al., 2014) at a Finnish software company show that software developers do not feel good about taking TD since they have to deal with it in the future. Finally, Peters (Peters, 2014) suggests that the presence of TD has a negative influence on developers' motivation. Even though none of the previous studies directly refers to the relationship between TD and developers' self-worth, this observation may be explained in terms of developers' tendency to produce high-quality software (Lim et al., 2012). In particular, since the occurrence of TD could hinder developers from achieving this goal, it may have a negative influence on developers' self-worth.

Additionally, our results suggest that TD management has a positive influence on the antecedents of morale, and consequently, it can be said that proper management of TD might increase the developers' morale. McConnell (2010) makes a similar argument in a blog post by suggesting that repaying TD "can be motivational and good for team morale." Finally, in their empirical study and based on the results of two surveys answered by 37 software professionals, Spinola et al. 2013) suggest that repaying TD may have a positive influence on morale; however, future research is needed to clarify this relationship. Damian and Chisan (2006) also suggest that continuous software improvements, in general, have a positive influence on developers' pride and morale.

Finally, in this study, we examine the relationship between TD and waste. Our result indicates that the more time that is wasted due to experiencing TD, the more the respondents feel that the progress is hindered by TD, which is also supported by the interview data, and it is therefore quite plausible. However, another interpretation is that the respondents with lower levels of progress tend to estimate the amount of waste to be higher. This may be explained in terms of confirmation bias whereby individuals tend to report information which confirms their viewpoints or beliefs. In other words, since developers believe that TD hinders their progress, it appears that they tend to estimate the amount of waste to be higher than its actual amount.

On the other hand, our results indicate a lack of correlations between the perceived waste due to TD and the interpersonal and affective dimensions of morale (ST1, ST4, ST5, ST7, and ST8). In fact, we would expect such associations to show up if the TD was explicitly accrued by one member of the team (or at least an employee somewhat close to the team), while the interest would be paid by another. However, due to potential high turnover and the presence of external consultancy or heavy outsourcing, it is likely that developers find themselves dealing with the waste generated by code written by someone who is unknown to them. This could potentially explain why TD waste does not seem to be associated with interpersonal or affective factors.

On the other hand, it is still somewhat interesting to notice that our results do not indicate an association between the TD waste and ST2, ST6 (future/goal dimension). In fact, we might have expected a relationship between the amount of TD waste and the confidence in taking on TD (ST2): the more waste is paid, the less one might feel confident in taking TD. However, this can potentially be explained by the fact that accruing TD alone does not generate waste: the waste could be caused by a lack of dealing with such TD after the accrual (lack of refactoring). As for ST6, one might expect the satisfaction of paying back TD to be associated with higher waste or lower waste (avoiding a high amount of waste by repaying the debt could generate more satisfaction). However, the two variables are not directly correlated, as one might pay back TD, while at the same time suffering from high waste from other TD items, which cannot be refactored by that specific person.

In general, while for ST3, there is a correlation between the presence of TD waste and lack of progress, the other variables cannot directly be linked to any form of generic TD waste, but only in specific circumstances. To further understand such associations, we would need to conduct specific studies where variables related to the affective state of participants are controlled or at least better known.

Based on our findings, it seems that developers' perception of the amount of time wasted on managing TD could trigger a vicious circle where the presence of TD leads to higher perceived waste, which continuously lowers developers' morale and productivity. However, it is important to acknowledge that the correlational analysis used when answering the research questions do not infer any causal relationships between the different studied variables and thus does not suggest any casualties.

This is in line with Tom et al. (2013), who argue that TD lowers developers' morale, which in turn increases the amount of technical debt. This phenomenon can be explained by Hardy's (Hardy, 2010) argument that "*the consequences of morale feedback into the antecedents. This feedback loop seems to act as a form of confirmation bias whereby information which confirms the prevailing morale state is selected and that which refutes it rejected*." Therefore, it can be said that low morale appears to be associated with a perceived lack of achievement and negative self-assessment (Barnett et al., 1999; Maslach et al., 2001). Thus, in the context of our study, when developers perceive TD to hinder their progress, their morale appears to declines and this, in turn, appears to cause them to assess their accomplishments lower and their waste higher. In other words, developers with low morale appear to have an over-exaggerated perception of the extent to which TD hinders their progress and the amount of time wasted on servicing TD. This, in turn, may lead to negative self-assessment (i.e., the de-

veloper experiences more waste), which lowers their morale even more, which consequently could worsen their perception of the time wasted due to experiencing TD (i.e., negative self-assessment).

### 5.1. Implications

Our study has several novel contributions for both software engineering research and practice. First, our study is the first that specifically concentrates on investigating the influence of TD on developers' morale and its relation to developer productivity. In doing so, we explored the impacts of TD and its management on the antecedents of morale. As a result, we were able to show that the occurrence of TD can reduce developers' morale, while TD management increases their morale. These findings are very encouraging since they clarify the relations between different dimensions of morale and TD. In particular, by emphasizing the importance of affective and future/goal dimensions of morale, this study suggests that, while making trade-offs leading to the occurrence of TD, software firms must better consider the short- and long-term consequences of TD on developers' behavior and productivity as well as on software development and maintenance costs and success. This becomes even more important considering the interviewees' controversial viewpoints about the necessity of repaying different types of TD items. For instance, some of the interviewees considered fixing "cosmetic" issues (e.g., documentation TD) to be pointless, while repaying architectural TD and testing TD to be very important since long-term costs and complexities associated with them can surpass the short-term benefits gained by taking on such TD. On the other hand, the results of this study suggest that software firms need to consider TD management more seriously, by investing more resources and promoting a high-quality culture in which developers are encouraged and rewarded for identifying and repaying TD. Showing appreciation and recognition is a good method for encouraging developers to repay TD since it seems to create a sense of satisfaction for developers. In particular, we argue that if developers are supported and encouraged to repay TD, they feel more satisfied with their achievements. In general, we could infer that a company culture that is encouraging TD repayment also increases team morale.

Our suggestion is supported by the results of an empirical study showing that a lack of remuneration and reward schemes reduces developers' morale and productivity (Foulds and West, 2007). This need becomes more obvious considering previous research suggesting that the level of employees' morale is correlated with their productivity (Damian and Chisan, 2006), (Fairley and Willshire, 2005), (Foulds and West, 2007), and project success (Hall et al., 2007).

Based on the results of this study, we propose a twofold model for the description of the relationship between TD and morale, as illustrated in Fig. 7. Each entity represents a variable that we have studied. Each arrow is starting from an entity, E1, and ending with another entity, E2, denotes a positive influence of E1 over E2. Each arrow is the result of our findings, as reported in the figure's legend and represents a correlational relationship between the entities.

This model shows the overall picture of our findings and how various variables interact with each other. Other relationships might exist, which are not observable from the data collected in this study and are not shown in our model.

As shown in Fig. 7, TD management promotes a culture within organizations in which developers are supported and appreciated for managing their TD. Promoting such a culture not only helps to prevent TD but also has a positive influence on the team morale itself, which ultimately increases developers' morale. On the other hand, the appropriate use of tools and methods (which need to be put in place by TD management) helps to prevent TD and, as a result, reduces the amount of waste of time generated by it. Limiting the amount of working time wasted because of TD has a positive influence on the sense of progress among developers, which increases both the team's morale and developers' morale.

In conclusion, as can be seen in Fig. 7, an appropriate level of TD management leads to a virtuous cycle (C-F-G-I-B2) for which the right culture and TD prevention mechanisms reinforce each other, leading to happier developers.

## 6. Limitations and threats to validity

As empirical research, this study is subject to different threats. We discuss these limitations according to four types of threats to validity suggested by Runeson and Höst (2009).

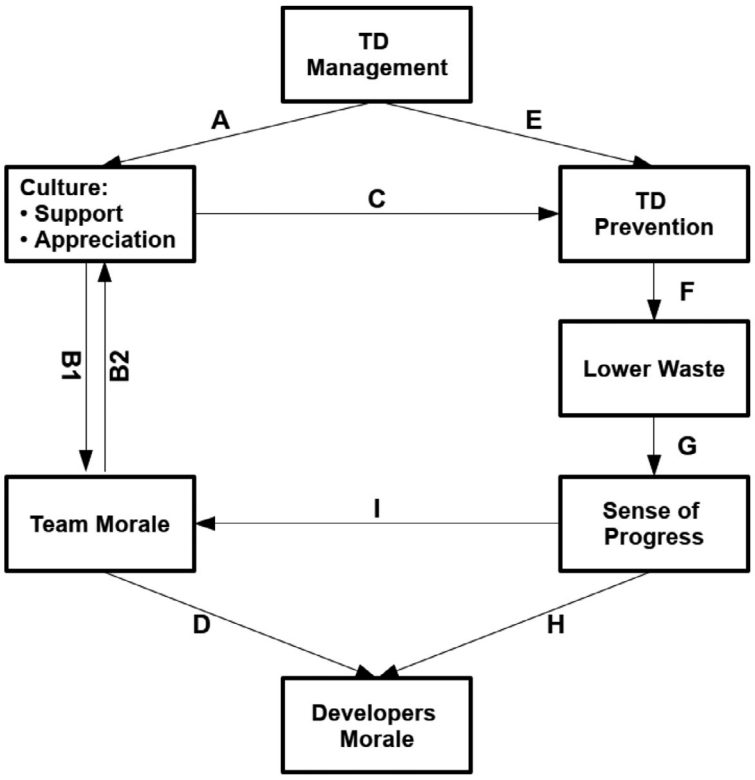
Concerning the *construct validity* (Runeson and Höst, 2009), there is a concern about using the right operational measures for studying the concept of morale. To mitigate this threat, we decided to follow an approach that relies on capturing different levels of morale by investigating its antecedent factors (Foulds and West, 2007). As a result, to avoid any misconception, instead of directly asking questions about morale, we asked questions about the influence of TD and its management on a set of factors affecting morale. Also, to avoid misunderstanding the research questions, both the interview and survey questionnaires were tested before data collection. Finally, we recorded all the interviews, and the research team reviewed the transcription of these interviewees to avoid any misinterpretation of the collected data. However, since in this study, we took a holistic approach to examine the influence of TD and its management on morale, future research is needed to provide a more specific understanding of how different types of TD and TD management activities can influence developers' morale.

Furthermore, even if there are other studies investigating the relation between other human factor constructs and productivity, such as, for example, practitioners' happiness (Graziotin et al., 2018), satisfaction, and motivation, this study focuses only on the relationship between productivity and morale as a human factor. However, in future studies, it would be interesting to compare our results with other human factors constructs.

This study also uses the amount of reported wasted working time due to experiencing TD as a proxy for productivity, but we do acknowledge that lack of waste does not guarantee productivity by default even if having to spend work time on TD tasks and activities is one obvious impediment towards achieving productivity. We also acknowledge a potential threat to validity in terms of respondents not having the conceptual clarity or the subjective ability to identify the correct amount of waste during the reporting occasions.

Regarding *internal validity* (Runeson and Höst, 2009), there is a risk that developers' morale is influenced by uncontrolled factors other than the occurrence and management of TD. To mitigate this risk, at the beginning of each interview, we asked the interviewees to choose specific items from their TD backlog and answer our questions accordingly. However, since we cannot be sure that no other uncontrolled factor has affected developers' morale, future research, preferably in a controlled experimental environment, is needed to address this limitation. Further, in this study, we have analyzed data to find a correlation between specific parameters in the reported data.

Further, we acknowledge that these statistical correlations do not imply causality between the variables. This is because there is a risk that the observed outcomes are affected by other factors that have not been accounted for in our empirical study. The results of this study could potentially be affected by this threat since some of our findings are correlational and potentially also indicate a causal relationship. For example, when we examine the amount of wasted time, correlated with the different statements. By performing this




| Arrow (positive impact) | Based on the results of |
|---|---|
| A | Section 4.1 & 4.2.1 |
| B1 & B2 | Section 4.1.3 & 4.2.1 |
| C | Section 5 |
| D | Section 4.2.1 |
| E | Section 4.1 |
| F | Section 4.2.2 |
| G | Section 4.1.2 & 4.2.1 |
| H | Section 4.1.2 & 4.2.1 |
| I | Section 4.1.2 & 4.2.1 |

**Fig. 7.** Twofold model of the relationship between TD and morale.

correlation, this validity could potentially have been violated by either finding relationships that are non-existent or missing real relationships that are wrongly deemed non-significant. However, by combining correlation analysis with analytical causality (from, e.g., the conducted interviews), causality links can be suggested. Therefore, to mitigate this threat, we triangulated the data by conducting follow-up interviews validating the derived results.

The *external aspect of validity* addresses the possibility of generalizing the findings (Runeson and Höst, 2009). The results in this study are based on data from several interviews, a survey, and a longitudinal study. The responses that our respondents gave in the longitudinal part of the study and during the survey and in the interviews might not be representative of the entire developer population. We cannot generalize the results. However, in some phases, we can rely on a good number of participating organizations (7) in different business and application domains. We also provide an online replication package at https://doi.org/10.5281/zenodo.3627885, with information and data organized per phase of the study (as presented in Fig 2), which will facilitate the replication of the study. Furthermore, in surveys, there is always a risk that the sample is biased, and, therefore, a potential threat relates to, for instance, the geographical, cultural, and demographic distribution of response samples.

Finally, since our findings are partially grounded in qualitative interviews, there is a concern about the *reliability* (Runeson and Höst, 2009) of the findings. We tried to mitigate this threat in different ways. First, we used triangulation both during the data collection and data analysis processes. To increase the reliability of the data collection process, at least two researchers were involved in planning and conducting the interviews (i.e., observer triangulation). Also, to complement our findings from the interviews, we conducted a longitudinal study and asked a group of software professionals to provide their opinions about a set of statements based on our results (i.e., methods triangulation), together with reporting on how much time they wasted. To increase the reliability of the findings further, even though the qualitative data analysis was performed mainly by the third author, each phase of the data analysis process and its outcomes were monitored and reviewed by the whole research team. Finally, while monitoring the responses to the surveys in the longitudinal study, we realized that one of the respondents had rated all the statements neutrally (i.e., straightlining). Therefore, we decided to remove the ratings of this respondent to avoid any threat to the validity of data analysis. It must be noted that excluding this respondent affects only the median value of the ratings for ST5.

Their anonymity was assured multiple times to the participants, both by email and in the introduction of the survey. However, we notice that for two statements in Table 5, for which a high rating could be perceived negatively from the perspective of interpersonal relations (i.e., ST1 and ST4) the median is '1', while for the more neutral statements the median represents a higher rating: this could mean that developers may have been reluctant to provide answers that might be perceived as 'hostile' to their colleagues. On the other hand, ST5 and ST7 (positive interpersonal relations) are not so extreme in the opposite direction, e.g., showing a very high median, but just 4 and 3, respectively. In conclusion, although a threat exists, we do not deem it to be extremely likely or disruptive.

Another limitation is related to the scope of our study. In our model in Fig. 7, we show how TD management essentially has only a positive impact on developers' morale and productivity, which in turn increases further the morale. However, this study has not investigated any negative impact of too much management to avoid TD being in place. In fact, industrial processes are routinely asking developers to perform maintenance tasks, refactoring activities, quality measurements might have the opposite effect on morale.

## 7. Conclusions

The presented research aimed to examine the potential impact of technical debt and its management on developers' morale and productivity. To the best of our knowledge, this study is the first study to assess these relationships empirically using a mixed-method approach.

The findings from this study make several contributions to the present TD research. First, the results from this study indicate that the occurrence of TD reduces developers' morale since the presence of TD hinders the developers' progress and reduces their confidence. Second, the results imply that proper management of TD increases developers' morale since it enables the developers to perform their tasks better and to improve software quality in the future. Thirdly, the results also suggest that proper TD management leads to a virtuous cycle where the right culture and TD prevention mechanisms reinforce each other, leading to less waste of time, followed by a continuous increase of the developers' morale and productivity.

Although the findings of this study are interesting and encouraging, future research can provide a better in-depth understanding of the relations between the occurrence and management of TD and developers' morale. Especially by using a wider range of data sources and utilizing different psychological measures, future research can perhaps indicate the strength of such impacts and explore any potential differences between development contexts. Further, even if our twofold model illustrating the relationship between TD and morale (Fig. 7) is an empirically grounded model, it can be further validated using, e.g., structural equation modeling with both surveys and software metric data as part of future research.

## Declaration of Competing Interests

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.

## CRediT authorship contribution statement

**Terese Besker:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Writing - review & editing, Visualization, Project administration. **Hadi Ghanbari:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Writing - original draft, Visualization. **Antonio Martini:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Writing - original draft, Visualization. **Jan Bosch:** Conceptualization, Methodology, Supervision.

## Acknowledgments

We thank all the anonymous interviewees and survey respondents for their contribution to this work.

## Appendix A. Codebook used for thematic analysis

| Themes | 2nd order themes | Codes for high morale | Sources[1] | References[2] |
|---|---|---|---|---|
| Affective antecedents | Valued & taken seriously | Appreciation | 6 | 8 |
| | | Trusted | 8 | 11 |
| | | Autonomy | 5 | 8 |
| | self-worth | Achievement | 8 | 14 |
| | | Feeling successful | 9 | 11 |
| | | Interesting work | 7 | 10 |
| | Support & communication | Good communication | 14 | 31 |
| | | Recognition | 8 | 13 |
| | | Leadership | 10 | 13 |
| | | Praise | 3 | 3 |
| Future/Goal antecedents | Vision of future | The attractiveness of vision | 1 | 1 |
| | | Clarity of vision | 11 | 20 |
| | | Better than present | 14 | 24 |
| | | Security | 1 | 1 |
| | | Challenge | 1 | 1 |
| | | Importance of the task | 13 | 28 |
| | Progress | Sense of progress | 11 | 21 |
| | | Feeling successful | 8 | 14 |
| | | Contribution to goal | 12 | 19 |
| Interpersonal antecedents | Influence of others | Contribution | 11 | 15 |
| | | Teamwork/pulling together | 12 | 18 |
| | | Pride | 3 | 3 |
| | | Interesting work | 0 | 0 |
| | Relationship with others | Cohesion | 9 | 15 |
| | | Good atmosphere | 11 | 20 |
| | | Helping others | 15 | 29 |
| **Themes** | **2nd order themes** | **Codes for low morale** | **Sources** | **References** |
| Affective antecedents | Valued & taken seriously | Marginalized | 2 | 3 |
| | | Injustice | 0 | 0 |
| | | Fragmentation | 0 | 0 |
| | self-worth | Boredom | 4 | 11 |
| | | Feeling failure | 8 | 17 |
| | Support & communication | Criticism | 7 | 11 |
| | | Being demanded | 4 | 9 |
| | | Lack of praise/recognition | 3 | 5 |
| Future/Goal antecedents | Vision of future | Lack of clarity | 4 | 4 |
| | | Changing objectives | 3 | 4 |
| | | Pointlessness | 4 | 7 |
| | | Lack of confidence | 7 | 11 |
| | | Future is seen as bleak | 6 | 11 |
| | | Insecurity | 1 | 1 |
| | Progress | Lack of progress | 8 | 21 |
| | | Interference from others | 6 | 10 |
| Interpersonal antecedents | Influence of others | Being dragged down by others | 5 | 7 |
| | | Bullying | 4 | 4 |
| | | Being demanded | 3 | 5 |
| | | Organizational politics | 3 | 6 |
| | Relationship with others | Isolation | 0 | 0 |
| | | Marginalization | 3 | 4 |
| | | Bad atmosphere | 0 | 0 |
| | | Division of workforce | 5 | 6 |
| **Themes** | **2nd order themes** | **Codes for TD and TD management** | **Sources** | **References** |
| Technical Debt | TD Occurrence | Causes of TD | 15 | 88 |
| | | Consequences of TD | 15 | 37 |
| | TD Management | TD identification | 15 | 59 |
| | | TD communication | 12 | 51 |
| | | TD measurement | 5 | 10 |
| | | TD monitoring | 8 | 11 |
| | | TD prevention | 10 | 24 |
| | | TD prioritization | 10 | 22 |
| | | TD repayment | 14 | 86 |
| | | TD representation-documentation | 10 | 26 |

1 - The column ***Sources*** shows the number of interviewees who discussed the focus of a code.
2 - The column ***References*** shows the number of quotes assigned to each code.
3 - The cells highlighted in green show that a majority of interviewees (> 50%) discussed the focus of a code.
4- The cells highlighted in orange show that almost half of the interviewees discussed the focus of a code.

## Appendix B. Interview protocol

**General Questions**

1. Could you please tell us a little bit about yourself, your team, and your company?
    a. Education
    b. Position and responsibilities
    c. Work experience in year
    d. Size of the team

**Violations of Quality Rules**

2. Do you follow any specific quality rules/coding standards in your company?
3. Are there any quality reviews to check if you comply with these quality rules? By whom?
4. How do you think these quality reviews (or use of Sonar) affect your work?
5. Can you show us an example of Technical Debt items that you have or you have had:
    a. Can you think of any specific reason for violating the quality rules in this case? (e.g., resource constrains, it was unintentionally or any other reasons)
    b. Do you remember who made the decision in this case? (Personal, team, management)
    c. How do you usually make decision in such situations? (e.g., decide yourself, discuss with others, or let others make the decision)

**Affective Antecedents**

6. In this specific case how (satisfied or confident) were you about your decision?
7. In general how (satisfied or confident) are you when violating quality rules?
8. Do you feel (upset / nervous) if someone else finds out that you have violated the rules?
9. Have you ever been somehow (criticized or punished) for violating these quality rules?
10. Have you ever been somehow (praised or rewarded) for fixing these issues?
11. Do you think that teammates and the management (appreciate) it when you fix these issues?

**Future/Goal antecedents**

12. How do you think violating these quality rules affects your work in future (are there any costs or difficulties)?
13. Do you think violating these rules really reduces the quality of software? Why?
14. Do you think it really (pays-off) to fix all these issues or it is (waste of time)? Why? (Which ones are the most important ones?)
15. How (important) do you think it is for managers to fix these issues?
16. Do you feel more (satisfied or confident) when you know these issues are fixed? Why?

**Interpersonal Antecedents**

17. From your perspective who should (take responsibility) for these issues? Why?
18. From your perspective who must (be responsible) for fixing these issues? Why?
19. How (fair) do you think it is if others violate the quality rules and then you have to fix them?

## Appendix C. Survey questions

1. How many years of experience in the Software Development area do you have?

- ◯ < 2 years
- ◯ 2 - 5 years
- ◯ 5 - 10 years
- ◯ > 10 years

3. What is your gender?

- ◯ Male
- ◯ Female
- ◯ Other / Don't want to share

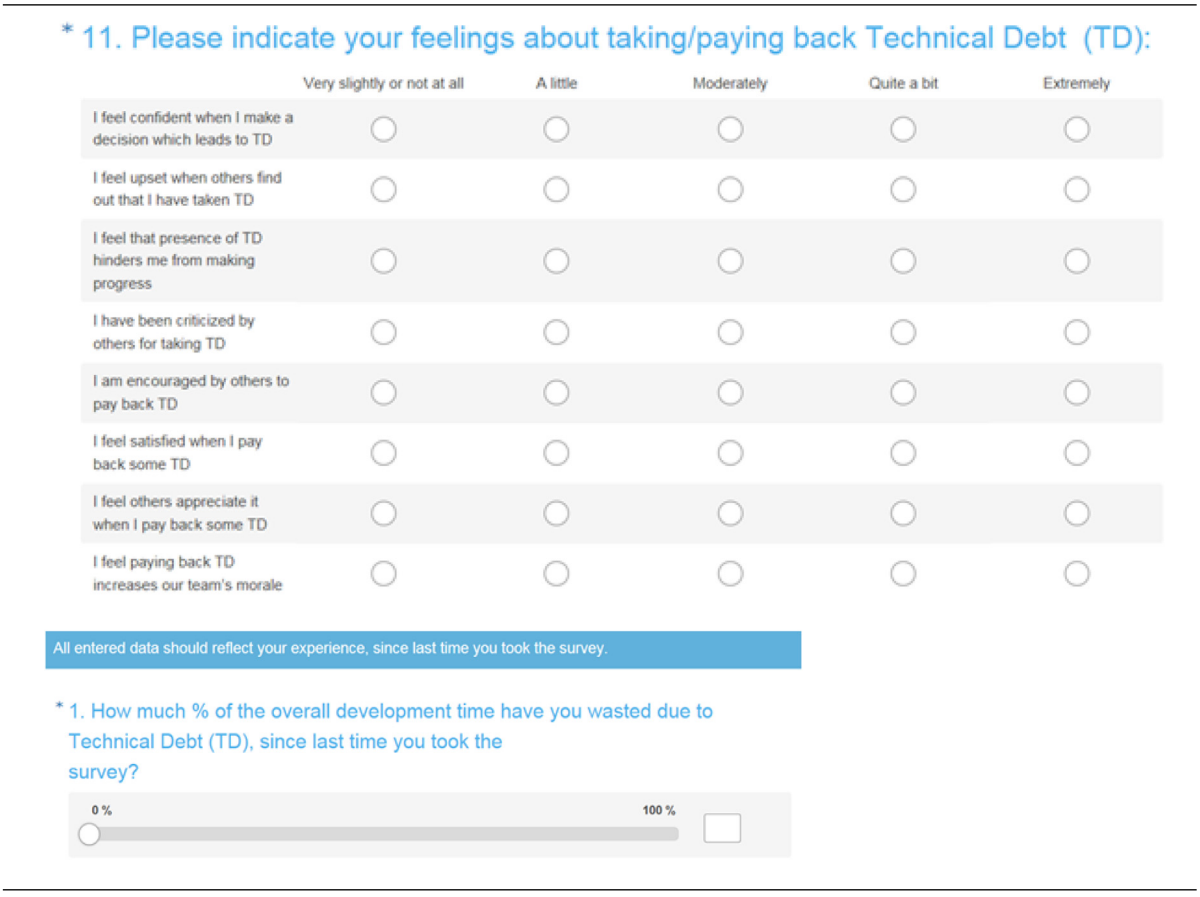
4. What is the highest level of education you have completed?

- ◯ No University education
- ◯ Bachelor degree
- ◯ Master degree (or Civilingenjör)
- ◯ Ph.D. degree

6. What type of software system are you developing?

- ☐ Real-time system
- ☐ Data management system
- ☐ Embedded system
- ☐ Web 2.0 / SaaS system
- ☐ Data analysis system
- ☐ Modeling and/or simulation system
- ☐ System Integration
- ☐ Other (please specify)

(continued on next page)




## Appendix D. Educational material

*What is Technical Debt?*

*Definitions*

The term Technical Debt was coined by Ward Cunningham: "Shipping first time code is like going into debt. A little debt speeds development so long as it is paid back promptly with a rewrite... The danger occurs when the debt is not repaid. Every minute spent on not-quite-right code counts as interest on that debt"

Steve McConnell's definition of technical debt has increasingly been accepted "A design or construction approach that's expedient in the short term but that creates a technical context in which the same work will cost more to do later than it would cost to do now (including increased cost over time)"

A shorter definition is: Technical Debt (TD) is a non-optimal solution in code (or other artifacts related to software development) that gives a short-term benefit, but cause a extra long-term cost during the software life-cycle.

*Terms in Technical Debt*

*Debt*: the sub-optimal solution implemented to achieve short-term benefits.

*Principal*: the cost of refactoring the sub-optimal solution.

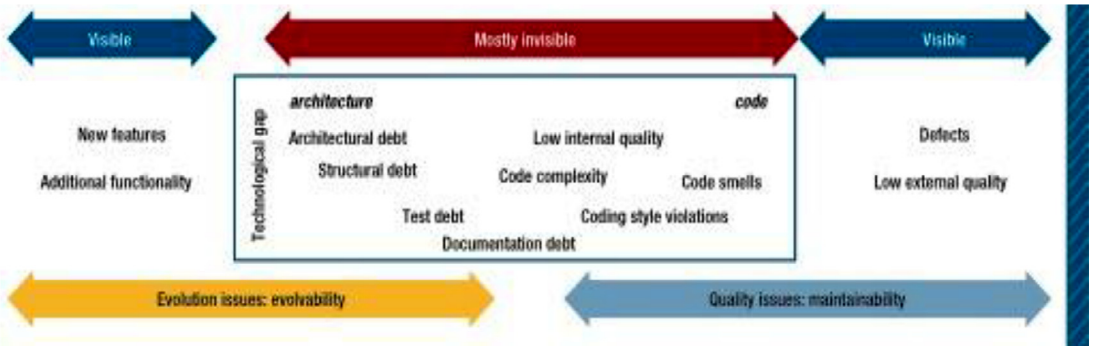
*Interest*: the extra-cost, current or estimated in the future, generated by the sub-optimal solution that would not be paid if the debt was not there.

Ideally, the Debt needs to be repaid when the *Interest* is at least greater than the *Principal*. In practice, the *Interest* needs to be much bigger than the *Principal*, otherwise it would not pay off.

Notice that the Interest mainly depends on the extra-costs that are going to occur when maintaining or evolving the system as well as issues that affect external quality and block new features. However, if the Debt generates a low interest (for example, that part of the code is or will be rarely changed), it is not convenient to refactor (repay) the Debt.

*Technical Debt Landscape*

Technical Debt is the invisible part in the middlebox: it is not bugs, it is not external quality, it is not lack of features or lack of functionalities (Kruchten et al., 2012).



## References


Ampatzoglou, A., Ampatzoglou, A., Avgeriou, P., Chatzigeorgiou, A., 2016. A financial approach for managing interest in technical debt. Lecture Notes Bus. Inf. Process. 257, 117–133.

Li, Z., Avgeriou, P., Liang, P., 2015. A systematic mapping study on technical debt and its management. J. Syst. Softw. 101, 193–220.

Besker, T., Martini, A., Bosch, J., 2018. Managing architectural technical debt: a unified model and systematic literature review. J. Syst. Softw. 135, 1–16 Supplement C.

Yli-Huumo, J., Maglyas, A., Smolander, K., 2014. The sources and approaches to management of technical debt: a case study of two product lines in a middle-size Finnish software company. In: Conf. Product-Focused Software Process Improvment, pp. 93–107.

Codabux, Z., Williams, B., 2013. Managing technical debt: an industrial case study. In: Proceedings of the 4th International Workshop on Managing Technical Debt, pp. 8–15.

Spinola, R.O., et al., May 2013. Investigating technical debt folklore: shedding some light on technical debt opinion. In: Managing Technical Debt (MTD), 2013 4th International Workshop on, pp. 1–7 20-202013.

Tom, E., Aurum, A., Vidgen, R., 2013. An exploration of technical debt. J. Syst. Softw. 86 (6), 1498–1516.

Fernández-Sánchez, C., Garbajosa, J., Yagüe, A., 2015. A framework to aid in decision making for technical debt management. In: 7th IEEE Workshop on Managing Technical Debt, pp. 69–76.

Peterson, C., Park, N., and Sweeney, P.J., 2008. “Group well-being: morale from a positive psychology perspective,” Appl. Psychol., vol. 57, no. s1, pp. 19–36.

McLeod, L., Doolin, B., 2012. Information systems development as situated socio-technical change: a process approach. Eur. J. Inf. Syst. 21 (2), 176–191 2012/03/01.

Feldt, R., Angelis, L., Torkar, R., Samuelsson, M., 2010. Links between the personalities, views and attitudes of software engineers. Inf. Softw. Technol. 52 (6), 611–624 2010/06/01/.

F. Fagerholm, and J. Münch, “Developer experience: concept and definition,” 2012 International Conference on Software and System Process (ICSSP), pp. 73–77.

Graziotin, X.Wang, Abrahamsson, P., 2014. Happy software developers solve problems better: psychological measurements in empirical software engineering. PeerJ 2, e289.

Abbott, J., 2003. Does employee satisfaction matter? a study to determine whether low employee morale affects customer satisfaction and profits in the business–to-business sector. J. Commun. Manage. 7 (4), 333–339.

Stowe, C.J., 2009. Incorporating morale into a classical agency model: implications for incentives, effort, and organization. Econ. Governance 10 (2).

Damian, D., Chisan, J., 2006. An empirical study of the complex relationships between requirements engineering processes and other processes that lead to payoffs in productivity, quality, and risk management. IEEE Trans. Softw. Eng., 32 (7), 433–453.

Fairley, R.E., Willshire, M.J., 2005. Iterative rework: the good, the bad, and the ugly. Computer 38 (9), 34–41.

Foulds, L.R., West, M., 2007. The productivity of large business information system development. Int. J. Bus. Inf. Syst. 2 (2), 162–181.

Hall, T., Jagielska, D., Baddoo, N., 2007. Motivating developer performance to improve project outcomes in a high maturity organization. Softw. Qual. J. 15 (4), 365–381.

Becker, C., Walker, D., McCord, C., May 2017. Intertemporal choice: decision making and time in software engineering. In: 2017 IEEE/ACM 10th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), 23–23, pp. 23–29 2017.

Cunningham, W., 1992. The wycash portfolio management system. In: 7th International Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA ’92), pp. 29–30 5–10 October.

Ghanbari, H., Besker, T., Martini, A., Bosch, J., 2017. Looking for peace of mind? manage your (Technical) debt - An Exploratory field study,. 11th International Symposium On Empirical Engineering and Measurement (ESEM).

Brown, N., et al., 2010. Managing technical debt in software-reliant systems. In: Proceedings of the FSE/SDP workshop on Future of software engineering research, pp. 47–52.

Martini, A., Bosch, J., Chaudron, M., 2015. Investigating architectural technical debt accumulation and refactoring over time: a multiple-case study. Inf. Softw. Technol. 67, 237–253.

Fagerholm, F., et al., 2015. Performance alignment work: how software developers experience the continuous adaptation of team performance in lean and agile environments. Inf. Softw. Technol. 64, 132–147.

Ralph, P., Kelly, P., 2014. The dimensions of software engineering success. In: Proceedings of the 36th International Conference on Software Engineering (ICSE), pp. 24–35.

Beecham, S., Baddoo, N., Hall, T., Robinson, H., Sharp, H., 2008. Motivation in software engineering: a systematic literature review. Inf Softw Technol 50 (9–10), 860–878 8//.

Verner, J.M., Babar, M.A., Cerpa, N., Hall, T., Beecham, S., 2014. Factors that motivate software engineering teams: a four country empirical study. J. Syst. Softw. 92, 115–127 2014/06/01/.

Laribee, D., 2009. Code cleanup - Using Agile techniques to pay back technical debt. Code Cleanup - Using Agile Techniques to Pay Back Technical Debt. https://msdn.microsoft.com/en-us/magazine/ee819135.aspx.

Vogel-Heuser, B., Neumann, E.-M., 2017. Adapting the concept of technical debt to software of automated production systems focusing on fault handling, mode of operation and safety aspects. IFAC-Papers OnLine 50 (1), 5887–5894 2017/07/01/.

Keyes, J., 2011. Social software engineering. Auerbach Series.

Tamburri, D.A., Kruchten, P., Lago, P., Van Vliet, H., 2013. What is social debt in software engineering? In: 2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2013 - Proceedings, pp. 93–96.

Alfayez, R., Behnamghader, P., Srisopha, K., Boehm, B., 2018. An exploratory study on the influence of developers in technical debt. In: IEEE/ACM International Conference on Technical Debt (TechDebt), pp. 1–10.

Salamea, M.J., Farré, C., 2019. Influence of developer factors on code quality: a data study. In: 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), pp. 120–125.

Hardy, B., 2010. PhD diss. University of Cambridge.

França, A.C.C., Gouveia, T.B., Santos, P.C.F., Santana, C.A., d. Silva, F.Q.B., 2011. Motivation in software engineering: a systematic review update. In: 15th Annual Conference on Evaluation & Assessment in Software Engineering (EASE 2011), pp. 154–163.

S. McConnell. "Technical debt. 10x software development [cited 2010 June 14]," 2019-10-07; http://www.construx.com/10x_Software_Development/Technical_Debt/.

Peters, L., 2014. Technical debt: the ultimate antipattern - The Biggest costs may be hidden, widespread, and long term. In: Managing Technical Debt (MTD), 2014 Sixth International Workshop on, pp. 8–10.

Lavallée, M., Robillard, P.N., 2012. The impacts of software process improvement on developers: a systematic review. In: 2012 34th International Conference on Software Engineering (ICSE), pp. 113–122.

Jaktman, C.B., 1998. The influence of organisational factors on the success and quality of a product-line architecture. In: Australian Software Engineering Conference (Cat. No.98EX233), pp. 2–11.

Lim, E., Taksande, N., Seaman, C., 2012. A balancing act: what software practitioners have to say about technical debt. IEEE Softw. 29 (6), 22–27.

Suryanarayana, G., Samarthyam, G., Sharma, T., 2015. Chapter 1 - Technical Debt. In: Suryanarayana, G., Samarthyam, G., Sharma, T. (Eds.), Refactoring For Software Design Smells. Boston: Morgan Kaufmann, pp. 1–7.

C.F. Evans Data Corp, and Stripe research, “The developer coefficient software engineering efficiency and its $3 trillion impact on global gdp.,” https://stripe.com/files/reports/the-developer-coefficient.pdf, 2018.

Ozkaya, I., 2019. The voice of the developer. IEEE Softw. 36 (5), 3–5.

A. Aldaeej, Towards effective technical debt decision making in software startups: association for computing machinery, 2019.

Giardino, C., Paternoster, N., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P., 2016. Software development in startup companies: the greenfield startup model. IEEE Trans. Softw. Eng. 42 (6), 585–604.

Rasch, R.H., Tosi, H.L., 1992. Factors affecting software developers’ performance: an integrated approach. MIS Quarterly 16 (3), 395–413.

Alahyari, H., Gorschek, T., Berntsson Svensson, R., 2019. An exploratory study of waste in software development organizations using agile or lean approaches: a multiple case study at 14 organizations. Inf. Softw. Technol. 105, 78–94.

Graziotin, D., Fagerholm, F., Wang, X., Abrahamsson, P., 2018. What happens when software developers are (un)happy. J. Syst. Softw. 140, 32–47.

Sadowski, C., Zimmermann, T., 2019. Rethinking productivity in software engineering. Apress.

Scacchi, W., 1991. Understanding software productivity. Int. J. Softw. Eng. Knowl. Eng. 1 (3), 293–321.

Oliveira, E., Viana, D., Cristo, M., Conte, T., 2017. How have software engineering researchers been measuring software productivity? a systematic mapping study. In: Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS), 2, pp. 76–87.

Ampatzoglou, A., Ampatzoglou, A., Chatzigeorgiou, A., Avgeriou, P., 2015. The financial aspect of managing technical debt: a systematic literature review. Inf. Softw. Technol. 64, 52.

Jensen, R.W., 2014. Improving software development productivity. Effective Leadership and Quantitative Methods in Software Management. Prentice Hall Press.

Sedano, T., Ralph, P., e. Péraire, C., 2017. Software development waste. In: Proceedings of the 39th International Conference on Software Engineering, Buenos Aires, Argentina, pp. 130–140.

Besker, T., Martini, A., Bosch, J., 2019. Software developer productivity loss due to technical debt—A replication and extension study examining developers’ development work. J. Syst. Softw. 156, 41–61.

Maslach, C., Schaufeli, W.B., Leiter, M.P., 2001. Job burnout. Annu. Rev. Psychol. 52, 397–422. doi:10.1146/annurev.psych.52.1.397.

Ployhart, R.E., Vandenberg, R.J., 2009. Longitudinal research: the theory, design, and analysis of change. J. Manage. 36 (1), 94–120.

Seaman, C.B., 1999. Qualitative methods in empirical studies of software engineering. IEEE Trans. Softw. Eng. 25 (4), 557–572.

Braun, V., Clarke, V., 2006. Using thematic analysis in psychology. Qual. Res. Psychol. 3 (2), 77–101.

Vaismoradi, M., Turunen, H., Bondas, T., 2013. Content analysis and thematic analysis: implications for conducting a qualitative descriptive study. Nurs. Health Sci. 15 (3), 398–405.

Czaja, R., Blair, J., 2005. Designing surveys: a Guide to Decisions and Procedures. Calif: Pine Forge Press, Thousand Oaks.

Wohlin, C., et al., 2000. Experimentation in Software engineering: an Introduction. Kluwer Academic Publishers.

Morrison, D.F., 1970. The optimal spacing of repeated measurements. Biometrics 26, 281–290.
Wickham, H., 2009. ggplot2: Elegant Graphics For Data Analysis. Springer Publishing Company, Springer-Verlag New York Incorporated.
Ghanbari, H., Vartiainen, T., Siponen, M., 2018. Omission of quality software development practices: a systematic literature review. ACM Comput. Surv. 51 (2).
Barnett, R.C., Brennan, R.T., Gareis, K.C., 1999. A closer look at the measurement of burnout. J. Appl. Biobehav. Res. 4 (2), 65–78.
Runeson, P., Höst, M., 2009. Guidelines for conducting and reporting case study research in software engineering. Empir. Softw. Eng. 14 (2), 131–164.
Kruchten, P., Nord, R.L., Ozkaya, I., 2012. Technical debt: from metaphor to theory and practice. Softw. IEEE 29 (6), 18–21.

**Terese Besker** is a Ph.D. candidate in the Software Engineering at Chalmers University of Technology, Sweden. She is working in the research fields of technical debt management. Before becoming a Ph.D. student, she had worked as a senior software engineer in the software industry for more than fifteen years. She also has a bachelor's degree in software engineering and a master's degree in applied IT from Chalmers. She has published several peer-reviewed articles in journals, conference and workshop proceedings.

**Hadi Ghanbari** is a postdoctoral researcher at the Department of Information and Service Management, Aalto University, Finland. He received his Ph.D. in Computer Science from the University of Jyväskylä, Finland (2017), and received M.*Sc*. in Information Systems from the University of Oulu, Finland (2012). He also holds a B.Eng. in Software Engineering (2005). dr.. Ghanbari conducts research mainly in the areas of information systems development and digital innovation. During his academic career, he has been involved in several industry-driven R&D projects both as a project manager and a researcher. Prior to his academic career, he has worked in several positions in the IT industry for more than eight years.

**Antonio Martini** is an Associate Professor at the Department of Informatics, software engineering group, at the university of oslo, Norway. Antonio has worked as a developer in the industry, carrying out various roles. Antonio is leading a project within a research consortium of academia and large international companies on the management of Technical Debt. Antonio also works with software architecture, agile software development and software measurements.

**Jan Bosch** is professor of software engineering and director of the Software Center (www.software-center.se) at chalmers university technology in sweden. Earlier, he worked as Vice President Engineering Process at Intuit Inc where he also led Intuit's Open Innovation effort s and headed the central mobile technologies team. Before Intuit, he was head of the Software and Application Technologies Laboratory at Nokia Research Center, Finland. Prior to joining Nokia, he headed the software engineering research group at the University of Groningen, The Netherlands. He received a MSc degree from the University of Twente, The Netherlands, and a Ph.D. degree from Lund University, Sweden. His-research activities include evidence-based development, software architecture, innovation experiment systems, compositional software engineering, software ecosystems, software product families and software variability management.