

**This is an electronic reprint of the original article.  
This reprint *may differ* from the original in pagination and typographic detail.**

**Author(s):** Ghanbari, Hadi; Besker, Terese; Martini, Antonio; Bosch, Jan

**Title:** Looking for Peace of Mind? Manage your (Technical) Debt : An Exploratory Field Study

**Year:** 2017

**Version:**

**Please cite the original version:**

Ghanbari, H., Besker, T., Martini, A., & Bosch, J. (2017). Looking for Peace of Mind? Manage your (Technical) Debt : An Exploratory Field Study. In ESEM 2017 : ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (pp. 384-393). IEEE Computer Society Press.  
<https://doi.org/10.1109/ESEM.2017.53>

All material supplied via JYX is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

# Looking for Peace of Mind? Manage your (Technical) Debt

## An Exploratory Field Study

Hadi Ghanbari

Faculty of Information Technology  
University of Jyväskylä  
Jyväskylä, Finland  
hadi.ghanbari@jyu.fi

Terese Besker, Antonio Martini, Jan Bosch

Software Engineering Division  
Chalmers University of Technology  
Göteborg, Sweden  
{besker, antonio.martini}@chalmers.se, jan@janbosch.com

**Abstract— Background:** In the last two decades Technical Debt (TD) has received a considerable amount of attention from software engineering research and practice. Recently, a small group of studies suggests that, in addition to its technical and economic consequences, TD can affect developers' psychological states and morale. However, until now there has been a lack of empirical research clarifying such influences. **Aims:** In this study, we aim at taking the first step in filling this gap by investigating the potential impacts of TD and its management on developers' morale. **Method:** Drawing from previous literature on morale, we decided to explore the influence of TD and its management on three dimensions of morale called *affective*, *future/goal*, and *interpersonal* antecedents. In so doing, we conducted an exploratory field study and collected data from software professionals active in different industrial domains through eight qualitative interviews and an online survey (n=33). **Results:** Our results indicate that TD mainly has a negative influence on future/goal and affective antecedents of morale. This is mainly because the occurrence of TD hinders developers from performing their tasks and achieving their goals. TD management, on the other hand, has a positive influence on all the three dimensions of morale since it is associated with positive feelings and interpersonal feedback as well as a sense of progress. **Conclusions:** According to the results of this empirical study, the occurrence of TD reduces developers' morale, while its management increases developers' morale.

**Keywords-Technical Debt; Morale; Behavioral Software Engineering; Exploratory Field Study; Mixed-method**

### I. INTRODUCTION

In recent years, the metaphor of Technical Debt (TD) has been a topic of interest within the software engineering discipline. Until now, the majority of previous studies has focused on investigating the technical, financial and organizational aspects of TD [1-3]. However, understanding and explaining, the role of individual developers in the occurrence of TD as well as the consequences of TD for them has received minimal attention in the literature. In recent years, several studies have suggested that TD has a negative influence on developers' emotions and affects [4-6] and their morale [7, 8]. However, none of these previous studies specifically focuses on investigating and explaining the relationship between TD and developers' morale.

Based on a multivocal literature review, Tom et al. [7] suggest that morale, alongside with quality, productivity, and

project risk, are the four main areas which are negatively influenced by the occurrence of TD. In particular, since the occurrence of TD reduces software quality, developers must spend more time and effort to improve the quality of software in future. As a result, developers' productivity will decrease while maintenance costs will increase in the long term [7, 8]. On the other hand, some studies suggest that developers do not feel comfortable about taking on TD [4, 5] and it lowers their motivation [6].

Software development is a sociotechnical phenomenon [9], and therefore, its success depends on both its social and technical aspects [10]. The findings from recent research indicate that positive affective states, such as happiness, satisfaction, and motivation, increase software developers' productivity and software quality [11, 12]. On the other hand previous studies, mainly from the management science [13, 14], have shown that there is a link between employees' morale and their productivity. However, there is a lack of studies focusing on investigating the relationship between TD and morale.

Considering the large number of previous studies which argue that software firms take on TD to boost their productivity and obtain added business value [2, 7], it becomes apparent that there is a need for empirical studies to explore and explain the effects of TD on developers' morale and ultimately their productivity. In this study, we aimed at taking the first step in addressing this gap by answering the following research questions:

- *RQ1: How does the occurrence of TD influence developers' morale?*
- *RQ2: How does the management of TD influence developers' morale?*

To answer these questions, we conducted a field study to explore the potential influence of TD, as well as its management, on developers' morale. Collecting data through eight interviews, complemented by an online survey (n=33), and drawing from morale literature [15], we propose that the occurrence of TD can reduce developers' morale, while proper management of TD increases developers' morale.

Our study has several novel contributions for 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.

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 behavioral and organizational consequences of TD more seriously. Additionally, since previous studies have shown the link between morale and developers’ productivity, our study indicates the need for future studies to further investigate the consequences of TD on developers’ morale and other psychological dimensions. Finally, based on the previous literature on morale [15], we introduce a novel approach for studying morale within the software engineering discipline. This approach, which relies on investigating the antecedent factors of morale, enables future research to better explore and measure software developers’ morale in different contexts.

## II. BACKGROUND

### A. Technical Debt

In recent years, TD has been widely studied in the software engineering literature [2, 7, 16, 17]. TD is composed of a *debt*, which is a sub-optimal technical solution that leads to a short-term benefit as well as to the future payment of an *interest*, which is the extra-cost due to the presence of TD (e.g., slow feature development or low quality) [3]. 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 literature and in software projects, the interest might largely surpass the gain, for example by leading to development crises [18].

There are several kinds of TD, such as Architecture TD, Testing TD, and Source Code TD [2], depending on what artefact and on what level of abstraction the sub-optimality has occurred. In recent years, there has been an increasing interest among software engineering researchers to provide novel solutions enabling software developers to manage TD. Management of TD consists of a set of activities which enable software development teams both to prevent the occurrence of TD and to deal with existing TD and its unwanted consequences [2]. It is worth noting that, in their systematic mapping study, Li et al. [2] classify TD management activities into identification, measurement, prioritization, prevention, monitoring, repayment, documentation and representation, and communication. However, discussing about each of these activities is out of the scope of this study.

### B. 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 [15]. 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 [15] shows that several concepts such as satisfaction, motivation, and happiness, have been used interchangeably to discuss morale. However, by conducting a longitudinal empirical study Hardy [15] clarifies that morale is different from these concepts. Therefore, in this study we use a

definition of morale provided by Peterson et al. [19] 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*”.

Previous studies have used different qualitative and quantitative methods for measuring morale, including direct (e.g., [20, 21]) and indirect measurement methods (e.g., [22, 23]). 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 [20]. 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 [15]. Consequently, Hardy [15] 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., confidence and happiness) while low morale is seen as a negative affective state [15]. The second category, *future/goal antecedents*, consists of factors related to the perception of individuals about the difference between future and today as well as their goals for future. For instance, believing in a better future is considered a key factor increasing morale [15]. Finally, the *interpersonal antecedents* of morale pertain to the factors related to the relationships and communication between individuals. For instance, having good relations with, or receiving positive feedback from, others contribute to higher levels of morale [15].

### C. Technical Debt and Morale

Tom et al. [7] argue that TD has a very strong negative effect on developers’ morale. Following this argument, we tried to identify previous studies that discuss about the relationship between TD and morale. In doing so, we were able to find only eight sources mentioning the potential influence of TD on developers’ morale or emotions (see Table I).

TABLE I. PREVIOUS STUDIES MENTIONING THE RELATIONSHIP BETWEEN TD AND DEVELOPERS’ MORALE OR EMOTIONS.

Ref	Argument	Source of Data
[5]	TD will hurt you later. It is better to pay upfront.	Interviews
[8]	Uncontrolled technical debt has a negative impact on the developers’ morale.	Literature
[24]	Engineers don’t like technical debt because they want to create perfect software.	Interviews
[25]	“Working off debt” can be motivational and good for team morale.	Opinion
	TD hurts developers personally [...] the developers suffer real consequences.	Comments on blog-post
[26]	TD reduces developers’ motivation.	Opinion
[6]	“Working off debt” can be motivational and good for team morale.	Literature
[7]	TD ultimately has negative impacts on morale.	Blog-post
	Developers would find the effects of technical debt frustrating in the long term.	Blog-post + Interviews
	A lot of the tasks to prevent or repay the accrual of TD aren’t very fun.	Interviews
[4]	Taking TD doesn’t feel good for developers.	Interviews

In particular, we found three articles [6-8] and one blog post [25] that have mentioned the influence of TD on developers' morale. In addition, during our literature search, we found four articles that mention the influence of TD on developers' emotions [4, 24, 26].

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 of these sources just mention the relationship between TD and developers' morale or emotions and investigating this phenomenon is not their primary research focus.

Therefore, we decided to conduct an exploratory field study to understand how TD, and possibly its management, could potentially influence software developers' morale. However, due to a lack of software engineering studies focusing on morale, we had to adapt a suitable approach from other disciplines. In doing so, we decided to utilize the indirect approach suggested by Hardy [15] to explore the potential influence of TD and its management on developers' morale (see Fig. 1).

As illustrated in Fig. 1 we try to understand how the occurrence of TD and its management may affect the three dimensions of morale suggested by Hardy [15]. 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 potential consequences of TD for developers and to understand which dimensions of morale are more influenced by TD.

### III. METHODOLOGY

To understand the potential impacts of TD and its management on developers' morale we collected data from software professional (i.e. developer, architect, and tester) active in different domains. In so doing, we followed a mixed method approach consisting of face-to-face interviews and an online questionnaire.

#### A. Interviews

We conducted three rounds of interviews during the summer and autumn 2016. In total, we conducted eight face-to-face interviews with software professionals from four different companies (see Table II). Company A develops equipment for aerospace crew and Company B is an international telecom company. Both of these companies are 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.

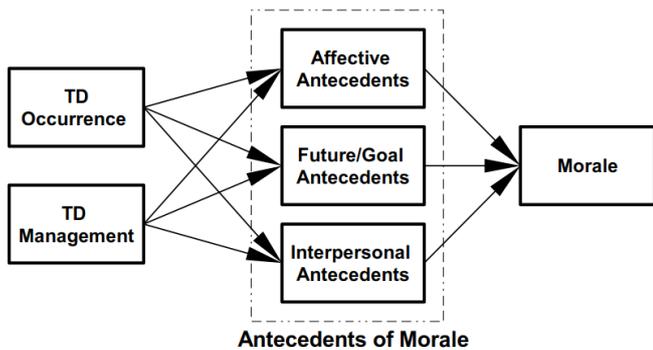


Fig. 1. Influence of TD and its management on antecedents of morale.

TABLE II. DEVELOPERS' FEELINGS TOWARDS TECHNICAL DEBT

ID	Role	Experience (year)	Company	Domain
I1	Software Developer	NA	Company A	Aerospace
I2	Software Developer	28		
I3	Software Developer	44		
I4	Software Developer	7	Company B	Telecom
I5	Software Developer	8		
I6	Software Developer	26		
I7	Testing Engineer	11	Company C	Healthcare
I8	Software Engineer	6	Company D	Aerospace

Finally, Company D is a German branch of an international enterprise providing software development and maintenance services to major aerospace customers such as European Space Agency.

We prepared a set of questions based on the antecedent factors of morale suggested by [15]. To improve the reliability of collected data at least two of the authors were participating in the interview sessions. In these interviews, when possible, we asked the interviewees to show us at least one specific item from their TD backlog<sup>1</sup> and answer our questions by considering that item. For example, we asked of the interviewees from Company A and Company D to pick some of the existing issues in *SonarQube* tool, while developers from Company B 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. All the interviews, which lasted 30 mins in average, were recorded and transcribed for further analysis.

#### B. Online survey

To complement our data and to further clarify the initial results from the interviews we decided to conduct an online survey. In doing so, a web-survey was designed and hosted by the online survey service called *SurveyMonkey*. The first draft of the survey was tested by the second author and one project manager to evaluate the understanding of the questions and the usage of common terms and expressions [27]. During this evaluation, we also monitored the time that was needed to answer the questionnaire.

The survey was made accessible between December 2016 and January 2017, and a reminder was sent out after two weeks to all the invited participants. The survey was anonymous, and the participation in the survey was voluntary. In designing the survey, we tried to form neutral questions and ordered 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 seven companies within our networks, all located in Scandinavia, with an extensive range of software development projects. In total, we received 33 valid responses.

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

<sup>1</sup> Company A and Company D use *SonarQube*—a tool used to identify Source Code TD, and Company B uses an in-house tool to identify ATD.

TABLE III. CHARACTERISTICS OF THE SAMPLE SURVEY

Individual		Company	
<b>Experience</b>		<b>Software system type(s)</b>	
< 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%
<b>Education</b>		Modeling and/or simul.	
Master degree	76,5%	2,9%	
Bachelor degree	20,6%	Data analysis sys.	
Ph.D. degree	2,9%	14,7%	
<b>Gender</b>			
Male	82,3%		
Female	17,7%		

The second part of the survey included questions based on our theoretical model presented in Fig. 1. In this part of the survey, 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.

### C. Data Analysis

To analyze the qualitative data collected from interviews we have used a thematic analysis approach [28]. 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 [28, 29]. Thematic analysis is especially suitable for studying the attitudes and behavior of people to explore a novel research phenomenon [29]. Since it has been suggested that a theoretical thematic analysis is more useful in testing an existing theory in a new context [29], we decided to perform a deductive thematic analysis [28] in three phases (see Fig. 2).

In phase 1, we first prepared a codebook based on two main sources that were identified during the literature review. First, based on [15], a set of codes related to morale were generated (see Appendix A). The second set of codes related to TD, and its management, were generated based on [2]. After preparing the codebook, the first author transcribed the recorded interviews and the research team reviewed the transcriptions in order to familiarize themselves with the data and to get an overall idea of the collected data. The interviews alongside with their transcriptions were added to a data analysis tool called *NVivo*.

In phase 2, the first author coded the interviews to identify data segments relevant to the research questions (see Fig. 3). Some of these initial codes were randomly picked and checked by the other authors, in order to triangulate the interpretation of the data and to minimize bias as much as possible.

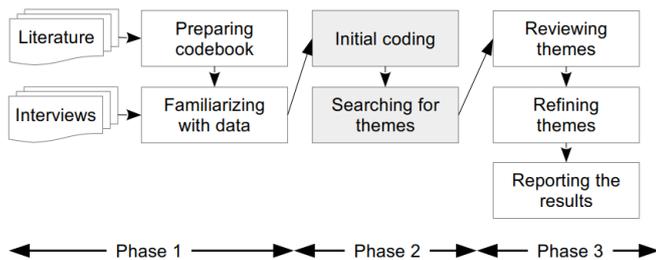


Fig. 2. Data analysis process.

“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 “Lack of progress”

We continued the data analysis process with 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 themes called “Progress” and “Consequences of TD”. By this, we were able to capture the relationship between themes to create our initial thematic map.

In Phase 3 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 first author continued to refine the codes and themes and the thematic map (see Fig. 4). During this phase we realized that the data from interviews do not support some of the original themes related to morale that were adapted from [15]. Therefore, we excluded those themes, which are shown in grey, from our final thematic map (see Fig. 4).

Finally we have prepared a discussion of the results of our data analysis to explain the relations between the occurrence and management of TD and 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. These propositions were then complemented by conducting an online questionnaire.

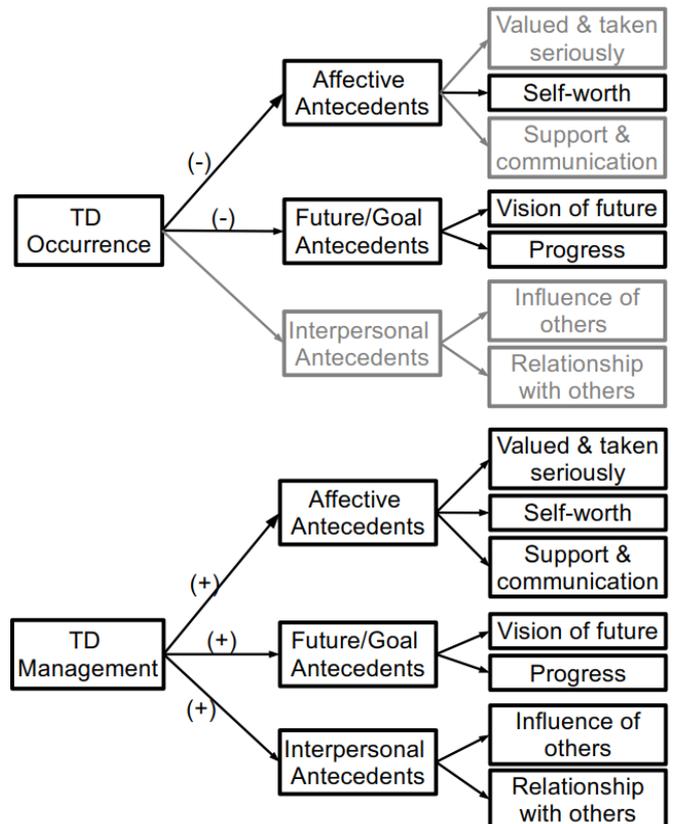


Fig. 4. Thematic map

## IV. RESULTS

### A. Results of the Interviews

The results from the interviews suggest that both presence and management of TD can influence developers' morale. In particular, the occurrence of TD mainly has a negative influence on the affective and future/goal antecedents of morale. On the other hand, the interview data indicate that management of TD has a strong positive influence on all the three dimensions of morale. In the following sections we discuss these findings in more detail.

#### 1) Influence on Affective antecedents

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*). From these second order themes, the occurrence of TD seems to have a negative influence only on developers' self-worth. In other words, it seems that the interviewees are used to the presence of TD and they do not have any strong affective reaction to its occurrence.

The majority 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 being criticized for introducing TD might be a bad feeling which makes it difficult to criticize others for taking TD. Thus, it is very important to know how to communicate with others and pointing out to their mistakes. Interviewee 3 raised this issue as shown in the following excerpt: "you would have to go and ask them why is it 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 the majority 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 personally 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 sense of insecurity and failure or being afraid and upset. For instance, Interviewee 2 mentioned this issue as: "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 have a tendency to perform a good job. On the other hand, some of the interviewees mentioned that working with software artefacts, especially legacy code, which contain 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<sup>2</sup>:

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

On the other hand, it seems that management of TD has an influence on 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 reasonable amount of autonomy to perform refactoring and re-paying TD. However, it seems that, from the interviewees' perspective, the companies do not reward TD management sufficiently. Even, one developer mentioned that in their company a slight praise or appreciation is missing, while another one said that in case developers cannot show its value and results, repaying TD is not appreciated by others.

On the other hand, the interview data indicates that managing TD is associated with a sense of self-worth among software developers. The majority of the interviewees think that identification and communication of TD enables them to improve their skills by understanding their mistakes and learning new aspects of software engineering. Interviewee 1 for instance mentioned that: "over the years [it has] being teaching me lots of stuff and made me a better developer in a shorter time than it would have been possible without it. I think it has been a huge change". In addition, it seems that managing TD leads to a sense of achievement and success among developers. Several interviewees mentioned that performing refactoring and re-paying TD is an interesting activity, which motivates them as well as enables them to constantly increase the quality of software artefacts and feel confident about their products.

However, several interviewees mentioned that, managing TD is not an easy and straightforward task. Especially, managing TD of legacy code is challenging since developers often need to perform a long chain of changes in different parts of the code until they actually get to fix the initial issue. Additionally, it is not easy to show the real value of such improvements and convince managers why it is necessary to perform them. Based on our data, it seems that fixing TD can be even dangerous in some cases since it might cause failure in a working system and others consider it as a result of unnecessary changes. Therefore, some developers might prefer not to touch the code and let the TD stay within the code.

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 as a task, which everyone can benefit from, and it is necessary for improving the quality of the software artefacts. Overall, based on these observations we propose that:

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

#### 2) Influence on Future/Goal antecedents

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. *vision of future* and *progress*). Based on the interview data, the influence of TD on the future/goal

<sup>2</sup> Please note that in acronyms used for propositions PTD stands for Proposition Technical Debt and PTDM stands for Proposition Technical Debt Management.

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 presence of TD hinders them from progressing in their tasks. At the same time it makes the future unclear since developers may face with unpredicted difficulties.

As it was mentioned by several interviewees, large amounts of TD makes the maintenance difficult and costly since understanding and working with software becomes problematic and there might be unpredictable issues. Additionally, 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 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 being aware of a large volume of TD in their artefacts or because it hinders them from performing their tasks. Several interviewees mentioned that introducing TD could make developers feel that they lack necessary skills and lower their confidence. As a result, they start to be more conscious about their decisions, and therefore, development speed slows down 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, which could have been avoided. For instance, *Interviewee 8* mentioned that: “*in the past, either us 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*”.

Additionally, several interviewees mentioned that when different modules are developed by different developers or teams, 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. The above-mentioned discussion leads us to propose:

- *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 present. These interviewees mentioned that they have a clear vision of how to properly manage TD within their company. In doing so, they use different techniques and tools to identify, document, communicate, prevent, and re-pay TD. Additionally, 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 re-paying TD to be necessary but for some items it is pointless. For instance, several

interviewees mentioned that re-paying architectural TD and testing TD is very important, while some of them mentioned that it is pointless to fix issues which are considered to be “cosmetic” (e.g. documentation TD) 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 re-paying 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 as a contribution to their teams' goals and by re-paying TD they feel successful in progressing towards such those goals. Such sense of progress is a positive feeling, which enables developers to perform their tasks easily and smoothly. Therefore, based on these observations we propose that:

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

### 3) Influence on Interpersonal antecedents

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., *influence of others* and *relationship 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 decisions have led to the occurrence of TD and it could be avoided by making better decisions. 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] has been written in a better way it would have been much faster and easier to implement this feature*”.

In regard to the relationship with others, it seems that from the interviewees' perspective there is a very good cohesion and understanding within their teams about the reasons behind the occurrence of TD. Most of them think that a combination of factors leads to the occurrence of TD, and therefore, they do not feel that previous developers dragged them down by introducing TD.

Since we could not discover strong evidence from our data indicating that TD has a negative influence on interpersonal antecedents of morale we propose:

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

Finally, concerning the influence of others, it seems that the majority of our interviewees consider TD management as a teamwork by which they contribute to a set of common goals within their teams. Especially, they consider conducting reviews as a positive contribution by which they are able to

identify TD and help each other in improving the quality of software. Interviewee 7 in the following excerpt refers to this matter: “the person who reviews my code is a very nice person so it is not like questioning me why did you write like this and if he says something it is just for making the code better”.

On the other hand, in regard to relationship with others, most of our interviewees think that the current team is responsible to manage TD and improve 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 re-paying 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 artefacts. Based on these observations we propose that:

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

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

### B. Results of the Online Survey

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 IV).

TABLE IV. STATEMENTS WHICH WERE RATED BY SURVEY RESPONDENTS

SID	Statements	Dimension of Morale
ST1	I have been criticized by others for taking TD.	Affective
ST2	I feel confident when I make a decision which leads to TD.	Affective
ST3	I feel that presence of TD hinders me from making progress.	Future/Goal
ST4	I feel upset when others find out that I have taken TD.	Interpersonal
ST5	I feel that others appreciate it when I pay back some TD.	Affective
ST6	I feel satisfied when I pay back some TD.	Future/Goal
ST7	I am encouraged by others to pay back TD.	Interpersonal
ST8	I feel that paying back TD increases our team’s morale.	All three

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 IV, 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. The summary of the ratings provided by the respondents is shown in Fig. 5.

We measured the central tendency and frequency of the respondents’ ratings (see Table V). 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. About dispersion, a lower Interquartile Range (IQR) value indicates higher levels of consensus among respondents while a higher IQR value implies lower consensus among respondents.

In regard to the influence of TD on the antecedents of morale, ST1 received very low rating (median = 1) with 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. In addition, ST2 received moderate rating (median = 3), which suggests that the respondents, with medium level of consensus (IQR = 2), do not feel very confident when introducing TD. These results indicate that the occurrence of TD has a negative influence on affective antecedents of morale. This observation supports PTD1, which was proposed based on the interview data.

Additionally, the ratings for ST3 (median = 3; IQR = 2) show that the survey respondents, with medium level of consensus, consider TD to moderately hinder them from making progress, which supports PTD2. Finally, ST4 received very low rating with high level of consensus among respondents (median = 1; IQR = 1). This indicates that the respondents’ consider the interpersonal consequences of TD to be very insignificant, which is in line with PTD3.

Concerning the impacts of TD management on antecedents of morale, ST5 received high rating (median = 4) with medium level of consensus among respondents (IQR = 2). This

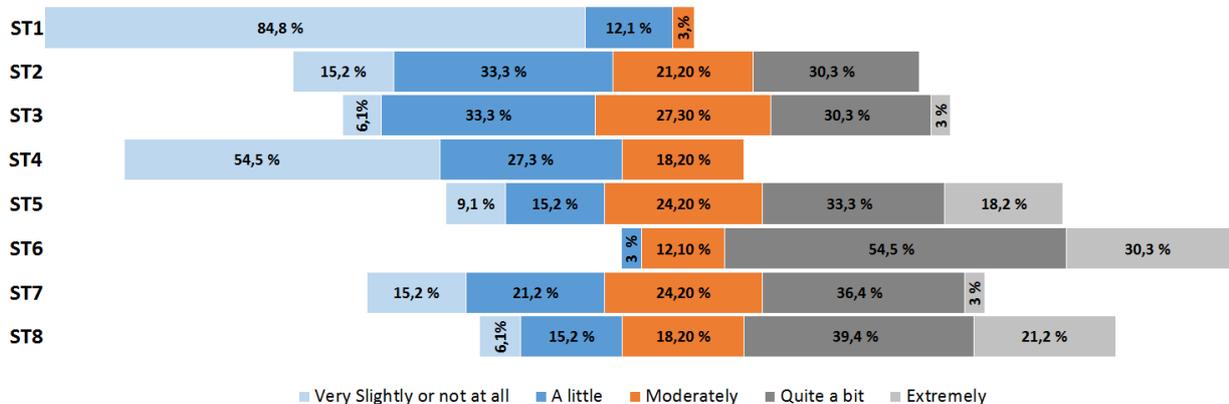


Fig. 5. Summary of the responses to the online survey.

TABLE V. MEDIAN AND IQR CALCULATED FOR SURVEY RESPONSES.

SID	Median	IQR	Supports the interviews	Related Proposition
ST1	1	0	Yes	PTD1
ST2	3	2	Yes	PTD1
ST3	3	2	Yes	PTD2
ST4	1	1	Yes	PTD3
ST5	4	2	Yes	PTDM1
ST6	4	1	Yes	PTDM2
ST7	3	2	Yes	PTDM3
ST8	4	1	Yes	PTDM1,PTDM2,PTDM3

<sup>a</sup> The values are based on the Likert scales where 1 = "Very slightly or not at all" and 5 = "Extremely".

observation supports PTDM1 and suggests that the survey respondents generally think that their efforts for managing TD are appreciated by their colleagues.

In addition, ST6 received high rating from respondents with high level of consensus (median = 4; IQR = 1), which indicates that the respondents strongly feel satisfied with re-paying TD. This supports PTDM2 which suggests managing TD has a positive influence on future/goal antecedents of morale since it is associated with a sense of progress. As it can be seen from Table V, ST7 received moderate rating (median = 3) with medium level of consensus (IQR = 2). This shows that the respondents think that they are moderately encouraged by others to re-pay TD, which indicates the positive influence of TD management on interpersonal antecedents of morale as proposed in PTDM3. Finally, ST8 was commonly rated as significant (median = 4; IQR = 1), which supports PTDM4 and suggests that in general management of TD has a strong positive influence on developers' morale.

As discussed earlier, some of the interviewees mentioned that developers' perception about 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, 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 that in general developers rarely criticize each other for the occurrence of TD.

In general, the results of the survey support our propositions that the occurrence of TD negatively influences the future/goal and affective antecedents of morale but not its interpersonal antecedents. On the other hand, management of TD positively affects all the three dimensions of morale.

## V. DISCUSSION

In this research, we aimed to answer two research questions. In response to RQ1, regarding the influence of TD on morale, it can be said that the occurrence of TD can reduce developers' morale mainly because it hinders their progress and makes it challenging to perform their tasks. In response to RQ2, it can be said that proper management of TD increases developers' morale since it is associated with positive personal

and interpersonal feedback and enables developers to better perform their tasks and to improve software quality in future.

### A. Implications

Our study has several novel contributions for both software engineering research and practice. First, our study is the first one, which specifically concentrates on investigating the influence of TD on developers' morale. In doing so, we explored the impacts of TD and its management on 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 as well as on software development and maintenance processes. This becomes even more important considering the interviewees' controversial viewpoints about the necessity of re-paying different types of TD items. For instance, the long-term costs and complexities associated with architectural or testing TD 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 quality culture in which developers are encouraged and rewarded for identifying and re-paying TD. This need becomes more obvious considering the link between morale and developers' productivity [11, 12]. Therefore, our study indicates the need for future studies to further investigate the consequences of TD on developers' morale and ultimately their productivity. Finally, exploring morale through its antecedent factors is a novel research approach within the software engineering discipline, which can be utilized in future research to better understand and explain the concept of morale within different software engineering contexts.

### B. Related work

Although in recent years considerable interest has been shown in conducting research on TD, there is a lack of studies exploring this phenomenon from individuals' perspective and particularly concentrating on explaining the behavioral 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 [4, 5, 24, 26] and morale [6-8, 25]. Therefore, we decided to conduct a field study to explore the potential impacts of TD, and its management, on developers' morale.

Even though Tom et al. [7] suggest that TD has "a strongly negative impact on morale", we argue that further empirical investigation is needed to precisely measure the strength of this negative impact. This is because our findings show that the occurrence of TD has a negative influence on the future/goal and affective antecedents of morale and not on its interpersonal antecedents. In other words, we believe 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 that suggest the presence of TD slows down development [5] and reduces developers'

productivity [7]. In addition, we were able to show that the occurrence of TD has a negative influence on affective antecedents of morale and in particular on developers' self-worth. In an exploratory field study and based on interviews with 35 software developers, Lim et al. [24] suggest that software developers have a negative attitude toward TD since they have a tendency to create "perfect software". The results of another case study conducted in a Finnish software company [4] show that software developers do not feel good about taking TD since they have to deal with it in future. Finally, Peters [26] suggests that the presence of TD has a negative influence on developers' motivation. Even though none of the previous studies directly refer to the relationship between TD and developers' self-worth, this observation can be explained in terms of developers' tendency of producing high quality software [24, 26]. In particular, since the occurrence of TD hinders developers from achieving this goal, it may have a negative influence on developers' self-worth.

Additionally, in this study we were able to show that TD management has a positive influence on antecedents of morale and consequently it can be said that proper management of TD increases developers' morale. McConnell [25] makes a similar argument in a blog post by suggesting that re-paying 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, Spínola et al. [6] suggest that re-paying TD may have a positive influence on morale, however, future research is needed to clarify this relationship.

### C. Limitations

As an empirical research, this study is subject to different threats. We discuss these limitations by according to four types of threats to validity suggested by Runeson and Höst [30].

Concerning the construct validity [30], there is a concern about using right operational measures for studying the concept of morale. To mitigate this threat, we decided to follow an approach, which relies on capturing different levels of morale through investigating its antecedent factors [15]. 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. In addition, to avoid misunderstanding the research questions, both the interview and survey questionnaires were tested prior to 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 examining the influence of TD and its management on morale, future research is needed to provide more specific understanding of how different types of TD and TD management activities can influence developers' morale.

In regard to internal validity [30], 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, are needed to address this limitation.

Concerning external validity [30], our results are mainly based on eight interviewees, and therefore, it is difficult to argue that our findings are generalizable. However, by collecting data from four different companies and conducting an online survey answered by 33 software professionals, we could show that our findings are relevant for other firms interested in considering TD and its management. Another potential threat refers to the demographic distribution of response samples, since most of the interview and survey data was collected in Scandinavia. Thus, the results might be different in other geographical and cultural areas.

Finally, since our findings are partially grounded in qualitative interviews, there is a concern about the reliability [30] of the findings. We tried to mitigate this threat in different ways. First of all, 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). Additionally, to complement our findings from the interviews, we conducted an online survey and asked a group of software professionals to provide their opinions about a set of statements based on our results (i.e. methods triangulation). However, further empirical studies, are needed to collect rich data and to provide a process-based explanation of how the influence of TD on morale changes over time during the development projects. To further increase the reliability of the findings, even though the data analysis was performed mainly by the first 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 online survey we realized that one of the respondents has rated all the statements neutrally (i.e., straight lining). Therefore, we decided to remove the ratings of this respondent to avoid any threat to validity of data analysis. It must be noted that excluding this respondent only affects the median value of the ratings for ST5.

## VI. CONCLUSIONS

Following a mixed-method approach and drawing from previous literature, we were able to show that the occurrence of TD can reduce developers' morale. On the other hand, the results clearly suggest that proper management of TD increases developers' morale. Although the findings of this exploratory field study are very 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 possibly indicate the strength of such impacts and explore any potential differences between development contexts.

## ACKNOWLEDGMENT

We thank all the anonymous interviewees, survey respondents, and reviewers for their valuable contribution to this work. H.G. thanks Faculty of IT, University of Jyväskylä for partially funding this research. We also thank the Software

Center at Chalmers University of Technology for partially funding this research.

## REFERENCES

- [1] A. Ampatzoglou, A. Ampatzoglou, A. Chatzigeorgiou and P. Avgeriou, "The financial aspect of managing technical debt: a systematic literature review," *Information and Software Technology* 64: 52-73, 2015
- [2] Z. Li, P. Avgeriou and P. Liang, "A systematic mapping study on technical debt and its management," *Journal of Systems and Software* 101: 193-220, 2015.
- [3] T. Besker, A. Martini and J. Bosch, "A systematic literature review and a unified model of ATD," *Euromicro Conf. on Software Engineering and Advanced Applications (SEAA)*, 2016, pp. 189-197.
- [4] J. Yli-Huumo, A. Maglyas and K. Smolander, "The sources and approaches to management of technical debt: a case study of two product lines in a middle-size finnish software company," *Conf. Product-Focused Software Process Improvement* 93-107, 2014.
- [5] Z. Codabux and B. Williams, "Managing technical debt: an industrial case study," *4th IEEE Workshop on Managing Technical Debt*, 8-15, 2013.
- [6] R.O. Spínola, N. Zazworka, A. Vetrò, C. Seaman and F. Shull, "Investigating technical debt folklore: Shedding some light on technical debt opinion," *4th IEEE Workshop on Managing Technical Debt*, 1-7, 2013.
- [7] E. Tom, A. Aurum and R. Vidgen, "An exploration of technical debt," *Journal of Systems and Software* 86(6): 1498-1516, 2013.
- [8] C. Fernández-Sánchez, J. Garbajosa and A. Yagüe, "A framework to aid in decision making for technical debt management," *7th IEEE Workshop on Managing Technical Debt*, 69-76, 2015.
- [9] L. McLeod and B. Doolin, "Information systems development as situated socio-technical change: a process approach," *European Journal of Information Systems* 21(2): 176-191, 2012.
- [10] R. Feldt, L. Angelis, R. Torkar and M. Samuelsson, "Links between the personalities, views and attitudes of software engineers," *Information and Software Technology* 52(6): 611-624, 2010.
- [11] D. Graziotin, X. Wang and P. Abrahamsson, "Do feelings matter? On the correlation of affects and the self-assessed productivity in software engineering," *Journal of Software: Evolution and Process* 27(7): 467-487, 2015.
- [12] D. Graziotin, X. Wang and P. Abrahamsson, "Happy software developers solve problems better: psychological measurements in empirical software engineering," *PeerJ*. 2: e289, 2014.
- [13] J. Abbott, "Does employee satisfaction matter? A study to determine whether low employee morale affects customer satisfaction and profits in the business-to-business sector," *Journal of Communication Management* 7(4): 333-339, 2003.
- [14] C.J. Stowe, "Incorporating morale into a classical agency model: implications for incentives, effort, and organization," *Economics of governance*. 10(2): 147-164, 2009.
- [15] Hardy, *Morale: definitions, dimensions and measurement*, PhD diss. University of Cambridge, 2010.
- [16] W. Cunningham, "The WyCash portfolio management system," in *Addendum to the proceedings on Object-oriented programming systems, languages, and applications (Addendum)*, British Columbia, Canada, 4: 29-30, 1992.
- [17] N. Brown, Y. Cai, Y. Guo, R. Kazman, M. Kim, P. Kruchten, E. Lim, A. MacCormack, R. Nord, I. Ozkaya and Others, "Managing technical debt in software-reliant systems," 47-52, 2010.
- [18] A. Martini, J. Bosch and M. Chaudron, "Investigating Architectural Technical Debt accumulation and refactoring over time: A multiple-case study," *Information and Software Technology*, 67: 237-253, 2015.
- [19] C. Peterson, N. Park and P.J. Sweeney, "'Group well-being: morale from a positive psychology perspective,'" *Appl. Psychol.*, vol. 57, no. s1, pp. 19-36.
- [20] P.D. Bliese and T.W. Britt, "Social support, group consensus and stressor-strain relationships: Social context matters," *J.Organ.Behav.*, vol. 22, no. 4, pp. 425-436.
- [21] L.K. Johnsrud, "'Maintaining morale: A guide to assessing the morale of midlevel administrators and faculty,'" 1996.
- [22] R.J. Vandenberg, H.A. Richardson and L.J. Eastman, "'The impact of high involvement work processes on organizational effectiveness a second-order latent variable approach,'" *Group & Organization Management*, vol. 24, no. 3, pp. 300-339.
- [23] D.A. Hibbs Jr and H. Locking, "'Wage dispersion and productive efficiency: Evidence for Sweden,'" *J.Labor Econ.*, vol. 18, no. 4, pp. 755-782.
- [24] E. Lim, N. Taksande and C. Seaman, "A balancing act: what software practitioners have to say about technical debt," *Journal of Software* 29(6): 22-27, 2012.
- [25] S. McConnell, "Technical Debt", 2007, [http://www.construx.com/10x\\_Software\\_Development/Technical\\_Debt/](http://www.construx.com/10x_Software_Development/Technical_Debt/).
- [26] L. Peters, "Technical Debt: the ultimate antipattern - the biggest costs may be hidden, widespread, and long term," *6th IEEE Workshop on Managing Technical Debt*, 8-10, 2014.
- [27] R. Czaja and J. Blair, "Designing surveys: a guide to decisions and procedures," Thousand Oaks, Calif: Pine Forge Press, 2005.
- [28] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative research in psychology*, 3(2): 77-101, 2006.
- [29] M. Vaismoradi, H. Turunen and T. Bondas, "Content analysis and thematic analysis: implications for conducting a qualitative descriptive study," *Nursing & Health Sciences* 15(3): 398-405, 2013.
- [30] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering* 14(2): 131-164, 2009.

## APPENDIX A THE ANTECEDENT FACTORS OF MORAL ADAPTED FROM HARDY [15]

<i>Themes</i>	<i>Second-order themes</i>	<i>Codes for high morale</i>	<i>Codes for low morale</i>
Affective Antecedents	Valued & taken seriously	Appreciation, Trusted, Autonomy	Marginalized, Injustice, Fragmentation
	Self-worth	Achievement, Feeling successful, Interesting work	Boredom, Feeling failure
	Support & communication	Good communications, Recognition, Leadership, Praise	Criticism, Being demanded, Lack of praise/recognition
Future/Goal Antecedents	Vision of future	Attractiveness of vision, Clarity of vision, Better than present, Security, Challenge, Importance of the task	Lack of clarity, Changing objectives, Pointlessness, Lack of confidence, Future is seen as bleak, Insecurity
	Progress	Sense of progress, Feeling successful, Contribution to goal	Lack of progress, Interference from others
Interpersonal Antecedents	Influence of others	Contribution, Teamwork/pulling together, Pride, Interesting work	Being dragged down by others, Bullying, Being demanded, Organizational politics
	Relationship with others	Cohesion, Good atmosphere, Helping others	Isolation, Marginalization, Bad atmosphere, Division of workforce