Replication Package for: *The Influence of Technical Debt on Software Developer Morale*

Publication information:

- Will be updated when study is published

Researchers:

- **Terese Besker** (corresponding author) - Chalmers University of Technology, Computer Science and Engineering, Software Engineering Gothenburg, Sweden, besker@chalmers.se
- **Hadi Ghanbari -** Department of Information and Service Management, Aalto University, Finland
- **Antonio Martini -** The University of Oslo, Programming and Software Engineering, Oslo, Norway
- **Jan Bosch -** Chalmers University of Technology, Gothenburg, Chalmers University of Technology, Computer Science and Engineering, Software Engineering, Gothenburg, Sweden

The replication package is composed of:

- **Methodology – section A**
- **Codebook used for thematic analysis – section B**
- **Interview Protocol - section C**
- **Survey Questions - section D**
- **Educational Material - section E**

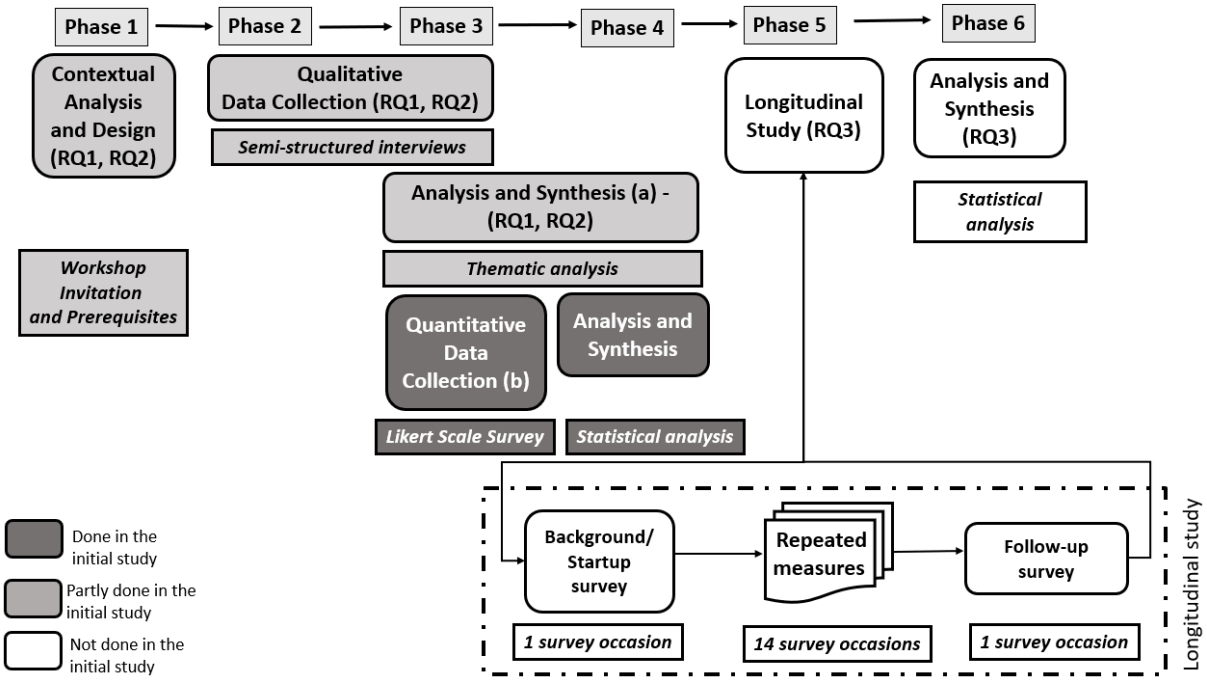
Replication Package – Section A

Methodology

The first section A in this Replication package describes the used methodology. For more details, please also look at the publication.

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.

As visualized in Figure below, 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 grey 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.



Visualization of the research design and research method used in each phase

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 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. To those 43 developers who approved to participate in the study, we emailed educational material (see Section E) intended to minimize inter-observer (all researchers communicate the same knowledge) and inter-instrument variability (all participants receive the same information).

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. Each interview lasted between 30 and 45 minutes, 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.

To improve the reliability of the collected data, at least two authors participated in the interviews.

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 and conduct the analysis in three phases:

In the first phase, we prepared a codebook based on two main sources that were initially identified.

Second, a set of codes related to TD occurrence and its management was generated. The full list of codes, second-order themes, and themes are shown in Section B. 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. Based on the feedback from reviewing the themes, the second author continued to refine the codes and themes and the thematic map. 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 conducted 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. During this evaluation, we also monitored the time that was needed to answer the questionnaire. The survey is shown in Section D.

The survey was anonymous, and participation in the survey was voluntary. The survey invitation was mailed directly to developers in companies within our networks.

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 below table, 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.

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

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.

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 survey is presented in Section E.

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 software developers

at 14 different survey occasions (i.e., twice a week for seven weeks). 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, 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 percent 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.

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 [65] version 1.1.1.

Replication Package – Section B
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
	Vision of future	Lack of clarity	4	4

Future/Goal antecedents		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

Replication Package – Section C

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?

Replication Package – Section D

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)

[ ]

* 11. Please indicate your feelings about taking/paying back Technical Debt (TD):

	Very slightly or not at all	A little	Moderately	Quite a bit	Extremely
I feel confident when I make a decision which leads to TD	○	○	○	○	○
I feel upset when others find out that I have taken TD	○	○	○	○	○
I feel that presence of TD hinders me from making progress	○	○	○	○	○
I have been criticized by others for taking TD	○	○	○	○	○
I am encouraged by others to pay back TD	○	○	○	○	○
I feel satisfied when I pay back some TD	○	○	○	○	○
I feel others appreciate it when I pay back some TD	○	○	○	○	○
I feel paying back TD increases our team's morale	○	○	○	○	○

All entered data should reflect your experience, since last time you took the survey.

* 1. How much % of the overall development time have you wasted due to Technical Debt (TD), since last time you took the survey?

0 % 100 %

Replication Package – Section E
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** [69].