The most common causes and effects of technical debt: first results from a global family of industrial surveys

Conference Paper · October 2018

DOI: 10.1145/3239235.3268917

Nicolli Rios
Federal University of Rio de Janeiro

Rodrigo Spínola
Universidade Salvador

Manoel Mendonça
Universidade Federal da Bahia

The Most Common Causes and Effects of Technical Debt: First Results from a Global Family of Industrial Surveys

Nicolli Rios
Dept. of Computer Science
Federal University of Bahia
Salvador, Brazil
nicollirioss@gmail.com

Rodrigo Oliveira Spínola
Graduate Program in Systems and Computer
Salvador University
Salvador, Brazil
rodrigo.spinola@unifacs.br

Manoel Mendonça
Dept. of Computer Science,
Federal University of Bahia
Fraunhofer Project Center
Salvador, Brazil
manoel.mendonca@ufba.br

Carolyn Seaman
Dept. of Information Systems
University of Maryland
Baltimore County
Baltimore, MD, United States
cseaman@umbc.edu

ABSTRACT

Background: The presence of technical debt (TD) brings risks to a software project and makes it difficult to manage. Several TD management strategies have been proposed, but considering actions that could explicitly prevent the insertion of TD in the first place and monitor its effects is not yet a common practice. Thus, while TD management is an important topic, it is also worthwhile to understand the causes that could lead development teams to incur different types of debt as well as the effects of their presence on software projects. *Aims:* The objective of this work is twofold. First, we investigate the state of practice in the TD area including the status quo, the causes that lead to TD occurrence, and the effects of existing TD. Second, we present the design of InsighTD, a globally distributed family of industrial surveys on causes and effects of TD, and the results of its first execution. *Method:* We designed the InsighTD in joint collaboration with several TD researchers. It is designed to run as an incremental large scale study based on continuous and independent replications of the questionnaire in different countries. *Results:* This paper presents the first results of the first execution of the survey. In total, 107 practitioners from the Brazilian software industry answered the questionnaire. Results indicate that there is a broad familiarity with the concept of TD. Deadlines, inappropriate planning, lack of knowledge, and lack of a well-defined process are among the top 10 cited and most likely causes that lead to the occurrence of TD. On the other side, low quality, delivery delay, low maintainability, rework and financial loss are among the top 10 most commonly cited and impactful effects of TD. *Conclusion:* With InsighTD, we intend to reduce the problem of isolated investigations in TD that are not yet representative and, thus, build a continuous and generalizable empirical basis for understanding practical problems and challenges of TD.

CCS CONCEPTS

• General and reference~Surveys and overviews • Software and its engineering

ESEM '18, October 11–12, 2018, Oulu, Finland

ACM ISBN 978-1-4503-5823-1/18/10…$15.00
https://doi.org/10.1145/3239235.3268917

KEYWORDS

technical debt, technical debt causes, technical debt effects, survey, family of surveys.

ACM Reference format:
Nicolli Rios, Rodrigo O. Spínola, Manoel Mendonça, and Carolyn Seaman. 2018. The Most Common Causes and Effects of Technical Debt: First Results from a Global Family of Industrial Surveys. In Proceedings of ACM/IEEE 12th International Symposium on Empirical Software Engineering and Measurement (ESEM'18). ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3239235.3268917

1 INTRODUCTION

Technical debt (TD) contextualizes the problem of pending development tasks as a type of debt that brings a short-term benefit to the project, usually in terms of increased productivity or less time to release software versions, but which may have to be paid with interest later in the development process [1, 2, 3]. It is common for a software project to incur debt during its development. However, its presence brings risks to the project and makes it difficult to manage [1, 6]. Several secondary studies have identified strategies that have been proposed to support TD management (TDM) [12]. All of these strategies are defined around a list of existing debt items to support decisions about whether and when the items should be paid. However, studying the underlying causes of TD, thus helping to identify actions that could prevent the TD items in the first place, is not yet a common practice. Neither is careful examination of the effects of TD, which would then inform prioritization strategies for paying off debt [3, 12, 16].

This is a point that deserves investigation because it is expected that TD prevention could be sometimes "cheaper" than TD repayment. Besides, when TD is prevented as much as possible, it also helps other TDM activities, and setting up TD prevention practices helps especially in catching inexperienced developers' 'not-so-good' solutions [16]. Knowing the causes for TD can support development teams in defining actions that could be taken to prevent the occurrence of debt items. From the effects perspective, implications of TD can affect projects in different ways. Having this information could aid in prioritization of TD items to pay off, by supporting a more precise impact analysis and also the definition of corrective actions to minimize possible negative consequences for the project.

Thus, while TDM is an important topic [6], it is also worthwhile to understand the motivations that could lead a development team to incur different types of TD and the implications of the presence of TD items in software projects, in other words TD causes and effects. Recently, some publications have tried to shed some light on this discussion [7, 8, 9, 10, 17, 18]. However, the existing evidence is still limited. Three of these are focused only on architectural TD [7, 17, 18] in detriment of the other types of debt. Ernst *et al.* [8] evaluated a predefined list of 14 causes of TD, which is limiting because this list does not necessarily represent the source of pain for the participants. Many of these studies [7, 9, 10, 17, 18] were very limited in scope, with a maximum of 17 participants from a maximum of 5 different organizations in each study. Ernst *et al.* [8] had a large number of participants, but they came from only three software organizations. The discussion around causes and effects of TD deserves a more comprehensive investigation so we can understand in a more generalizable way the reasons that lead software teams to incur debt and the pain that developers suffer because of its presence in their projects. Understanding TD from the practitioners' perspective is critical to guide research directions.

This paper presents the design and the initial results of the *InsighTD Survey – Investigating causes and implications of TD*. InsighTD is a globally distributed family of industrial surveys on TD. It is designed to run as an incremental large scale study based on continuous and independent replications of the questionnaire in different countries. Currently, researchers from five countries (Brazil, Finland, Netherlands, Serbia, and United States) have already joined the initiative. Its goal is to investigate the state of practice and industry trends in the TD area including the *status quo*, the causes that lead to TD occurrence, the effects of its existence, how these problems manifest themselves in the software development process, and how software development teams react when they are aware of the presence of debt items in their projects.

We present here the design of the family of surveys, as well as the initial results of its first execution in Brazil. In total, 107 practitioners from the Brazilian software industry answered the questionnaire between December/2017 and January/2018. Results show that there is a broad familiarity with the concept of TD. *Deadlines, inappropriate planning, lack of knowledge, and lack of a well-defined process* are among the top 10 cited and most likely causes that lead to the occurrence of TD. On the other side, low quality, delivery delay, low maintainability, rework, and financial loss are among the top 10 most commonly cited and impactful effects of TD.

The main contributions of this work are two-fold. First, the global family of surveys not only allows researchers to reproduce the results and their interpretation, but also allows practitioners to evaluate their own TD situation against overall industrial trends. Second, together, knowledge on causes and effects provide useful information to support decision makers on how to deal with TD. This set of information is still missing in the technical literature.

Besides this introduction, this paper is organized in seven more sections. Section 2 presents some related work. Next, Section 3 discusses the design of InsighTD and presents the current status of its execution. The results of its first execution are presented in Section 4. The implications of the study for researchers and practitioners, and the comparison to previous work are presented on Section 5. Then, we discuss some threats to validity in Section 6. Finally, Section 7 presents some final remarks and the next steps of this work.

2 RELATED WORK

Causes and effects of TD are still not well explored in the technical literature. This topic is rarely the central focus of any study, so in this section we describe a number of studies that looked at causes and effects of TD as a side issue, and concentrate on the analyses of those ancillary results.

Ernst *et al.* [8] reported the results of a survey of software engineers and architects working in software projects from two large multinational corporations and one government research lab, and addressed the following research questions: (i) do professional software engineers have a shared definition of technical debt?; (ii) are issues with architectural elements among the most significant sources of technical debt?, and (iii) are there practices and tools for managing technical debt? The results indicated that while participants believe the metaphor is itself important for communication, existing tools are not currently helpful in managing the details. Specifically talking about sources of TD, the authors asked participants to rank a randomly ordered predefined list of 14 choices with respect to the amount of debt they represent on their projects. The authors found that immature decisions about the design architecture are the key source of TD.

In other related work, Martini *et al.* [7] conducted a multiple-case embedded case study in seven sites at five large companies to investigate the current causes for the accumulation of architectural TD (ATD). The authors investigated two research questions: (i) what factors cause the accumulation of ATD?, and (ii) what are the current trends in practice in the accumulation and recovery of ATD over time? As one of the results, the authors provided a taxonomy of causes and their influence in the accumulation of ATD.

Still in the context of this same study, Martini and Bosch [17] investigated three additional research questions: (i) what are the most dangerous ATD items in terms of effort paid later?, (ii) what are the effects triggered by such ATD items?, and (iii) are there socio-technical patterns of events that trigger the creation of ATD leading to particularly dangerous interest to be paid? The authors reported that TD items can be contagious, causing other parts of the system to be contaminated with the same problem, which may lead to nonlinear growth of interest. The authors also presented a model of ATD effects that can be used for TD repayment prioritization.

This last topic was revisited in more detail in [18], where Martini and Bosch presented the results of a multiple case study involving six cases in four large companies to investigate two research questions: (i) what is the information needed by product owners and architects to prioritize ATD with respect to feature development? and (ii) what are the differences between architects and product owners when prioritizing ATD with respect to features? According to the authors, delivery time, maintenance costs and risk, would benefit greatly from information related to the effects of architecture debt. The authors also highlighted how measures of ATD effects, especially contagious debt, quality issues and "double" effort would be appreciated by software developers.

Yli-Huumo *et al.* performed several studies in industrial environments to investigate the role of TD in software development [9, 10]. In [9], the authors investigated the causes and effects of TD and what management and reduction strategies/practices are being used for TD. Through an exploratory case study with two independent software product lines in a mid-sized Finnish software company, they interviewed 12 practitioners with both business and development background. The results indicated that the primary reasons for incurring TD were management decisions that were made during the project to reach deadlines, or unknowingly due to lack of knowledge. In the long term, if TD is not paid back, it may generate quality issues in the software, which will later show as economic losses, such as extra work and decreased productivity.

In [10], Yli-Huumo *et al.* conducted semi-structured interviews with 17 representatives from two software organizations and concluded that workarounds (TD) are often intentional decisions and forced by time-to-market requirements. However, the stakeholders are not always familiar with the negative consequences of taking workarounds, like additional hours, costs, and poor quality.

As we can observe, there is already evidence in the technical literature that point out some causes and effects of TD on software projects. However, the existing evidence is still limited in several ways. Most obviously, the sample sizes tend to be quite small in most existing studies (see Table 1). The one with the largest sample [8] is still limited by the number of different organizations, and by the fact that a pre-defined list of TD causes, which constrained participants to fit their experience into this format. Further, half of the relevant studies focused on ATD, just one type of debt. Table 1 summarizes the main limitations of the related works.

Table 1: Related work limitations

	Cause/ Effect	Comprehensiveness		Representativeness	
		TD Type	Predefined causes/effects?	Sample size	# of organizations
[7]	C	ATD	N	-	5
[8]	C	General	Y	536	3
[9]	C/E	General	N	12	2
[10]	C/E	General	N	17	2
[17]	E	ATD	N	-	5
[18]	E	ATD	N	-	4

Understanding the causes and effects of TD from the developers' perspective is critical to guide research directions. The current state of the literature has stimulated us to deeply investigate these topics. To the best of our knowledge, the study presented in this paper is the first characterized as a large scale study in the TD area, involving researchers from different institutions around the world. Its design and the results from its first execution in Brazil will be discussed in the next sections.

3 A GLOBAL FAMILY OF SURVEYS ON TD

We present, in this section, the design of the family of surveys on TD (InsighTD). The design establishes the foundations for the survey to be continuously replicated in different countries. The idea behind these replications is to pursue generalizable results about the state of practice in the TD area including the status quo, the causes that lead to TD occurrence, the effects of its existence and how these problems manifest themselves in the software development process. The rationale behind choosing countries as scopes of replication is twofold: (i) organizing the work and making the dissemination of the survey wider, (ii) investigating whether differences in local development practices could influence how participants experience the TD concept.

By replicating the survey, we will incrementally put together data about practical problems of TD considering different development cultures, organization sizes, development methodologies, and so on. Each replication can bring several findings to the area. Further, the combination of data from different locations can shed some light on possible differences or similarities regarding perceptions of TD in specific development cultures.

To support the dissemination of results and collaboration among researchers, we will provide for each replication a shared survey infrastructure using the same questionnaire and a set of instruments to guide the data analysis. We also have a communication plan that defines how the results and achievements will be communicated among the participants of the project. Finally, we also have a website (available at td-survey.com) where further details about the project and news about its replications are continuously updated.

In the following, we discuss our research questions and the methodology design. Then, we define the target population, the questionnaire itself, and the data analysis procedures.

3.1 Research Questions

The overall design of InsighTD is based on the four research questions presented in Table 2. The goal of RQ1 is to investigate how disseminated among professionals the concept of TD is. RQ2 aims at identifying possible causes and the causes most likely to lead to the occurrence of TD. RQ3 aims at identifying possible effects that the presence of debt items have on software projects, and the effects that have a bigger impact. Finally, RQ4 investigates how software development teams react (e.g. monitor, pay off, or define preventive actions) when debt items are identified in their projects. Besides these questions, we also define some specific questions to characterize the survey participants and their respective organizations.

Table 2: Research Questions

RQ1	Are software professionals familiar with the concept of TD?
RQ2	What causes lead software development teams to incur TD?
RQ3	What effects does TD have on software projects?
RQ4	How do software development teams react when they are aware of the presence of debt items in their projects?

3.2 Methodology

InsighTD has been planned cooperatively with several TD researchers. Its design comprises four stages (Fig. 1): conception, validation, initiation, and international replication. The design also establishes that there are activities performed in isolation in Brazil and activities that are executed in cooperation with international TD research communities.

The first two stages, which comprise the planning step of InsighTD, correspond to the activities of conception and validation of the survey planning and questionnaire. The third stage corresponds to the first execution of the survey, which has already been performed in Brazil, and its first replication. Based on the results of the first execution, we have a baseline report and a better definition on how we can execute the data analysis and the synthesis of the results.

After the first execution, the empirical package is now available for partners from other countries, so they can replicate the survey and reuse the whole set of instruments in the last stage. At the time of writing this paper, researchers from five countries (Brazil, Finland, Netherlands, Serbia, and United States) have already joined the project. The first replication in United States is about to begin. Each replication of the survey will be performed independently by different researchers in different countries. All of them will use the same survey infrastructure and version of the questionnaire. The result of the continuous replication of the survey will be a rich empirical dataset on

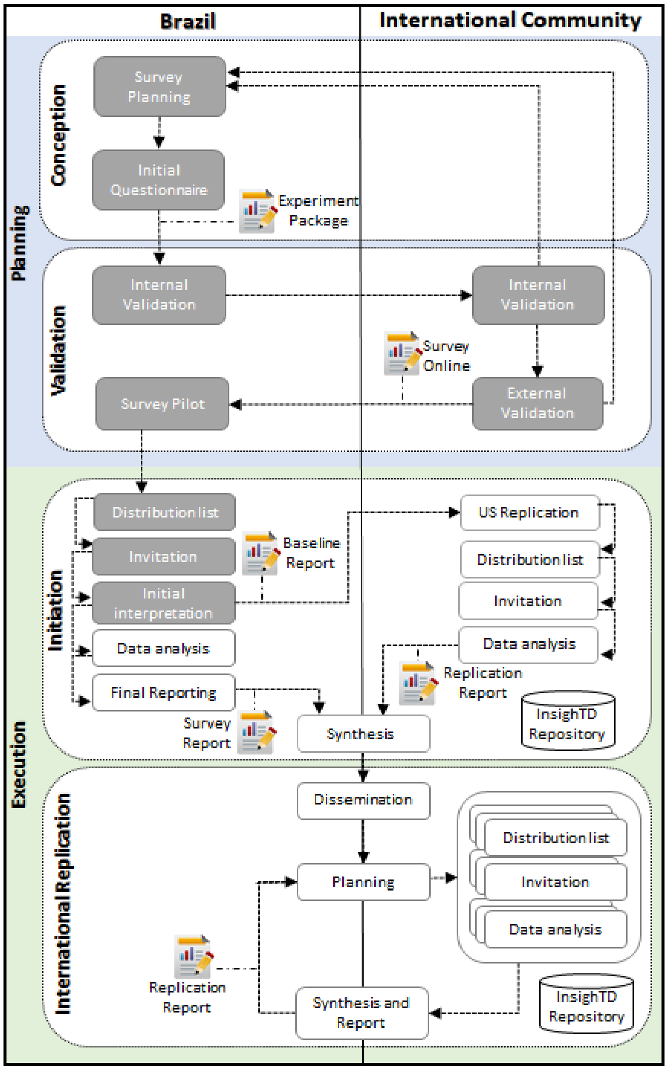

Figure 1: Study Design

causes and effects of TD that will be used by partners of the project to perform isolated or joint data analyses.

We describe the four stages of our methodology in more details in the next subsections.

3.2.1 Conception

This stage included: definition of the research questions, design of the family of surveys, elaboration of survey instruments, definition of target population, and initial discussion about data analysis. All these activities were carried out by the first two authors of the paper. The TD concepts and the decisions considered in this stage were grounded on two comprehensive literature reviews on TD: the mapping and tertiary studies reported in [3] and [12], respectively. Besides, we also considered some specific articles presenting survey studies and discussion on TD causes and effects [7, 8, 9, 10, 16, 17, 18]. Those articles, presented in the related work and discussion sections, supported the definition of the boundaries of the InsighTD project.

In this stage we also define the Core Team (CT) of InsighTD, that is responsible for executing the first round of the survey in Brazil and leading the project, and Replication Teams (RT). Table 3 shows who is currently involved in InsighTD with his/her respective role.

Table 3: Involved Researchers

Researcher	Review	Team
Nicolli Rios	-	CT-BR
Dr. Rodrigo Spínola	-	CT-BR
Dr. Manoel Mendonça	Internal *(plan. + questionnaire)*	CT-BR
Dr. Carolyn Seaman	Internal *(plan. + questionnaire)*	CT-US
Dr. Clemente Izurieta	-	RT-US
Dr. Davide Falessi	-	RT-US
Dr. Johannes Holvitie	Internal *(questionnaire)*	RT-FIN
Dr. Jesse Yli-Huumo	Internal *(questionnaire)*	RT-FIN
Dr. Paris Avgeriou	-	RT-NED
Dr. Vladimir Mandić	-	RT-SER
Dr. Nebojša Taušan	-	RT-SER
Robert Ramač	-	RT-SER
Dr. Forrest Shull	External*(plan. + questionnaire)*	-

3.2.2 Validation

The validation phase comprised three main steps: internal validation, external validation, and pilot study. The internal and external validations intend to ensure that the survey questions are clearly interpretable and sufficiently complete to answer the research questions, i.e., it should increase the internal and the construct validity. During these activities, we asked the reviewers to analyze if, for example: the study planning is well structured, the questions on the questionnaire could be used to answer the research questions, the questionnaire is clear and easily understood.

The validation activities were performed individually and incrementally. Thus, a second reviewer only started to review the instruments after the previous one had finished his/her task. In total, the team took about six months to complete this stage (including the pilot study execution).

The first internal validation was performed by a senior Brazilian industry consultant and researcher (the third author of this paper), who then joined the *core team*. In this first step, our goal was to have overall first impressions of the instruments. We had two onsite meetings to discuss and improve both the InsighTD planning and questionnaire. Further rounds of internal validation were performed by members of the international community: a senior US researcher (the fourth author of this paper, who also joined the core team), and two other researchers from Finland.

To perform the external validation, we invited an experienced researcher in the empirical software engineering and TD areas from the US. This researcher is not part of the InsighTD core or replication teams, acting as an independent reviewer of the instruments.

During internal and external validation, we received feedback concerning the design of the study, the research questions, and the questionnaire itself (e.g. clarity, ease of understanding, size). All feedback was discussed and considered by the core team. When appropriate, changes were made in the InsighTD study planning and questionnaire (e.g., adjustments on the used definition of TD, inclusion of the definition of traditional, hybrid, and agile process models). When not, we have discussed the suggestions with reviewers until we reached consensus.

For the pilot study, our goal was to observe if the questionnaire was well understood by a small number of participants who represent the target population of the study. We also intended to observe the effort involved to answer the questionnaire.

After internal and external validation, we implemented the survey using the Google Forms infrastructure and then conducted the pilot study. We used our personal contacts from the software industry to choose pilot participants, trying to select professionals with different levels of experience. In total, we invited five pilot participants, including one experienced developer with more than 10 years of experience, two midlevel software engineers with about 2-3 years of experience, and two inexperienced participants. After completing the questionnaire, the participants were asked to fill in a feedback form containing questions about how much time it took to complete the task (the mean time was about 20 minutes), impressions about questions (e.g. clarity, ease of understanding, size), and improvement points. Their feedback and the analysis of the responses served to identify vague questions and incomplete answer options in the closed questions, thus, increasing the internal and the external validity. Finally, after pilot execution, the members of the core team performed a last internal validation of the questionnaire.

3.2.3 Initiation

The initiation phase comprises two main steps: the administration of the survey in Brazil (completed, and reported in this paper) and, after, its first replication in United States (pending).

The initial activity of each survey execution is the creation of a distribution list, which contains the list of participants that will be invited to answer the questionnaire with their respective e-mail. More details on how the list was created for the initial survey in Brazil, and the target population is discussed in *Section 3.3 Population*.

The invitation to participate in the survey contained basic information about the goals of the survey and the link to the survey. We also described our target population and asked the participants to forward the invitation if possible. Finally, we explained the anonymous nature of the survey.

The first data analysis activity was of the results of the first execution in Brazil, and produced the baseline report. Once the baseline report was released, the first replication could begin. After data analysis is done for both the first execution in Brazil and the first replication in United States, we will conduct a first synthesis and use the results to further disseminate the survey among research communities. Thus, at the end of this phase, we will have a clear view of how to proceed with replications, data analyses, and also the synthesis of the results originating from different replications. The replication teams will then be equipped to go further and start the international replication of the survey.

3.2.4 International Replication

The last phase comprises the replication of the survey in other countries. Each replication will be performed independently by different researchers from different countries using the empirical package. As part of each replication, comparative analyses and investigation of country-specific results can be performed. To ensure a reproducible generalization and the openness of the results to the partners, the anonymized results will be available in the InsighTD repository. The overall goal is to establish a representative dataset to investigate industrial trends in TD.

3.3 Population

Since we intend to investigate the state of the practice, the questionnaire will be answered only by practitioners. Further, as several types of debt can affect software projects across different areas (e.g. test, requirements, code, design) [3], we also intend to search for professionals that work on a variety of roles in a software project.

For the first survey in Brazil, we used multiple strategies to reach the target population. We utilized social media in particular (LinkedIn), giving us direct access to a large number of professionals with whom we did not have previous contact. Specifically, LinkedIn allows the use of keywords to search for professionals with specific expertise. In InsighTD, we will search for professionals of a specific country (in the first instance, in Brazil) by type of debt. For example: to find professionals that could provide insightful answers about test debt, we used keywords like tester, test analyst, and test manager. The same applies to the other types of debt. Thus, the following keywords were considered: configuration analyst, configuration manager, developer, process analyst, product owner, programmer, project manager, requirements analyst, software architect, software engineer, system analyst, test analyst, test manager, and tester.

Besides social media, participants were also solicited from industry-affiliated member groups, mailing lists, and industry partners.

3.4 Questionnaire

The research questions guided the definition of the questionnaire, whose questions are summarized in Table 4 (the full questionnaire is available in the instruments section of the website td-survey.com). We also defined some specific questions to characterize the participants and their respective organizations. In total, we defined 28 questions. The questionnaire begins with characterization questions, followed by questions for RQ1, RQ2, RQ3, and RQ4. In Table 4, we also present the type of each question: multiple choice closed question or open question. Some of the closed questions include a free text option (e.g., other) so that the participants can express their opinion more appropriately.

The characterization questions (Q1 to Q8) ask participants about, for example, the size of his/her company, size of the system (in terms of LOC) he/she is working on, number of people involved in that project, participant's role, and her/his level of experience in that role. We also presented the definitions of agile [13, 15], hybrid [14, 15], and traditional [15] process models and asked the participants which one was followed by the development team.

After completing the characterization questions, we have some questions defined for RQ1 (Q9 to Q15). Initially, we ask (Q9) how familiar the participants are with the concept of TD. The available options are: "Never heard of it", "I have read about it in books / articles", "I have been on projects where I recognized TD but the project did not explicitly manage it" and "I have been on projects where we attempted to actively manage TD". Next, we ask participants to define TD in their own words. Then, we present a TD definition adapted from McConnell [11]: "*Technical debt contextualizes the problem of outstanding software development tasks (for example, tests planned but not executed, pending code refactoring, pending documentation update, use of bad design practices, code that does not exhibit good coding practices) as a kind of debt that brings a short-term benefit to the project (normally in terms of higher productivity or shorter release time of software versions), that may have to be paid later in the development process with interest (for example, a poorly designed class tends to be more difficult and costly to maintain than if it had been implemented good object-oriented practices)*". We chose to use this definition because it is already very disseminated in the technical literature and it is aligned with the types of debt identified by Alves *et al.* [3] and Rios *et al.* [12]. After, we ask (Q11) the participants how close this definition is to their understanding of the TD term, from the following options: "Very close", "Close", "Far", "Very far" and "Had no prior knowledge of TD".

We also ask (Q13) participants to provide an example TD item that occurred in their project (this example would then be used as the basis for answering questions about causes and effects) and, next, we asked (Q15) how representative that example was, using the following options: "It was a unique instance", "It is the type of thing that happens from time to time in the Project", and "It is the type of thing that happens very often in the Project".

For RQ2, questions Q16 to Q19 support the identification of the causes that lead development teams to insert debt items into their projects. Initially, referring to the example TD item cited by the participant in Q13, we asked the participant what led the development team to incur the TD in that example. Then, we asked if there were any other motives or reasons or causes that contributed to the occurrence of that instance of debt. We then asked this question one more time, in order to elicit more specific causes. The objective here was to dig further and further down, beyond the "obvious" causes, and also to find combinations of causes. Next, in Q19, we asked participants to list up to five causes he/she considered the ones that most likely lead to TD, considering all the cases of TD he/she has encountered.

Regarding RQ3, we defined questions Q20 and Q21 to identify effects of the presence of TD in software projects. Initially, in Q20, again referring to the example TD item cited by the participant in Q13, we asked the participant what impacts were felt in the project in that example. Next, in Q21, we asked participants to list up to five effects he/she considered highest impact, considering all the cases of TD he/she has encountered.

Finally, for RQ4 (Q22 to Q28), we intend to collect some data that helps us to understand how TD has been managed in practice, in particular with respect to prevention, repayment, and monitoring.

As we can observe in Table 4, open questions have an important role in our questionnaire (50% of the questions). As this is the first comprehensive study on causes and effects of TD, we decide to not present an initial list of causes and effects to avoid limiting the responses from the participants. We are more interested in hearing the voices of practitioners on their real problems than ask them if they agree or not with a predefined list that might not reflect their sources of pain.

In Section 4, we discuss an initial set of results from the survey in Brazil. In particular, results for the questions marked in gray in Table 4 will be discussed. Space does not permit a full account of all results, which will be shared in future publications. This analysis defines the baseline report of the study as defined in the overall InsighTD design.

3.5 Data Analysis

The survey instrument is composed of a mix of closed and open questions. Thus, we need to rely on a variety of procedures for data analysis.

For the analysis of the answers to closed questions, we first relied on descriptive statistics to get a better understanding of the data. We used the mode and median for the central tendency of the ordinal and interval data. For the nominal data, we calculated the share of participants choosing each option.

Table 4: Survey Questions (simplified)

RQ	No.	Question	Type
-	Q1	What is the size of your company?	Closed (SC)
	Q2	In which country you are currently working?	Closed (SC)
	Q3	What is the size of the system being developed in that project? (LOC)	Closed (SC)
	Q4	What is the total number of people of this project?	Closed (SC)
	Q5	What is the age of this system up to now or to when your involvement ended?	Closed (SC)
	Q6	To which project role are you assigned in this project?	Closed (SC)
	Q7	How do you rate your experience in this role (at the time)?	Closed (SC)
	Q8	Which of the following most closely describes the development process model you follow on this project?	Closed (SC)
RQ1	Q9	How familiar you are with the concept of Technical Debt?	Closed (SC)
	Q10	In your words, how would you define TD?	Open
	Q11	How close to the above TD definition is your understanding about TD?	Closed (SC)
	Q12	Are there any parts of the definition above from McConnell that you disagree with?	Open
	Q13	Please give an example of TD that had a significant impact on the project that you have chosen to tell us about:	Open
	Q14	Why did you select this example?	Open
	Q15	About this example, how representative it is?	Closed (SC)
RQ2	Q16	What was the immediate, or precipitating, cause of the example of TD you just described?	Open
	Q17	What other cause or factor contributed to the immediate cause you described above?	Open
	Q18	What other motives or reasons or causes contributed either directly or indirectly to the occurrence of the TD example?	Open
	Q19	Considering all the cases of TD you've encountered in different projects, and the causes of those TD cases, which causes would you say are the most likely to lead to TD (ordered by likelihood of causing TD)? Please list up to 5 causes.	Open
RQ3	Q20	Considering the TD item you described in question 13, what were the impacts felt in the project?	Open
	Q21	Considering all the cases of TD you've encountered in different projects and the effects of that TD that you have personally experienced, which 5 effects would you classify as the effects that have a bigger impact (ordered by their level of impact).	Open
RQ4	Q22	Do you think it would be possible to prevent the type of debt you described in question 13?	Closed (SC)
	Q23	If yes, how? If not, why?	Open
	Q24	Once identified, was the debt item monitored?	Closed (SC)
	Q25	If yes, how? If not, why?	Open
	Q26	Has the debt item been paid off (eliminated) from the project?	Closed (SC)
	Q27	If yes, how? If not, why?	Open
	Q28	Considering your personal experience with TD management, what actions have you performed to prevent its occurrence?	Open

To analyze answers given to the open questions on causes and effects of TD, we applied qualitative data analysis techniques [4] [5]. As the answers given for RQ2 and RQ3 were not related to any previous expectations, we followed an inductive logic approach to generate a new theory based on the given qualitative data. We applied manual coding on the RQ2 and RQ3 open questions as follows. Initially, the first and second authors individually coded the set of all answers for two subsets of related questions (RQ2: Q16+Q17+Q18+Q19, and RQ3: Q20+Q21). This involves open coding as described in [5], then axial coding to derive higher level categories. Next, they discussed possible differences in their coding until they reached consensus. Thus, we analyzed the data by attaching codes to small coherent units in the answers, and categorizing the emerged concepts (causes/effects) in a hierarchy of categories. This process was performed iteratively until reaching a state of saturation (the point where no new codes or categories were identified).

Thus, from an initial dataset of raw causes (Q16 to Q19) and effects (Q20 and Q21), we started to code (step 1) and, then group (step 2) them by similarity. For example, three participants cited the following causes in raw form: "Deliver the requested functionality in a tight deadline", "Short project time", and "The deadline forced the team to put the features in the systems, only ensuring that it would work". We initially coded these three chunks with "deadline", "short time", and "deadline", respectively. In step 3, we could identify these three example as different nomenclature for the same causes/effects. Then, we unified the names of sets of causes/effects using the most commonly used term in that subset, which was "deadline" in this example. After repeating these steps on the whole data sets, we had the final list of causes and effects.

4 RESULTS FROM BRAZIL

The survey in Brazil was online from December 7th, 2017, until January 7th, 2018. We sent the invitation to potential targets by e-mail or LinkedIn profile. In LinkedIn, we searched for software practitioners that work (or worked) in several software development areas (e.g., testing, coding, documentation, requirements, and management). In total, we sent the survey invitation to about 513 professionals and 112 of them completed the full questionnaire. This represents an approximate response rate of 22%. However, this is a rough estimate because social media and mailing lists do not allow accurate measurement of the number of individuals that read our recruitment message or read it but chose not to participate. Five participants who answered the questionnaire were excluded from the final dataset because they were not working in Brazil at that time (3) or did not provide a valid example of a TD item according to McConnell's TD definition in Q13 (2 - "I don´t know anything about TD" and "Errors that arise while fixing other issues").

Several types of expertise were represented in the collected data, as we can observe in Table 5. Most participants work as a developer, followed by manager, tester, software architect, and requirements analyst. When asked about their level of experience in their role, most participants indicated they are proficient (36%), followed by competent (32%), expert (20%), beginners (11%), and novice (1%).

Organizations of different sizes are represented in the dataset as we can see in Fig. 2. They are well distributed among small (28%), mid (45%), and large size (27%) companies. By observing the data in more detail, we can see that most participants (mode) work in organizations with more than 2,000 employees, closely followed by enterprises with 11-50 and 51-250 employees. The median size is 251-500 employees. Therefore, the participants tend to work in larger companies, but we have representatives from companies of all sizes.

Table 5: Participant Roles

Role	#	%
Developer	44	41%
Project Leader / Project Manager	14	13%
Test Manager / Tester	12	11%
Software Architect	10	9%
Requirements Analyst	10	9%
Process Analyst	4	4%
Infrastructure analyst	3	3%
DBA	3	3%
Performs multiple functions	3	3%
Business Analyst	2	2%
Configuration Manager	2	2%

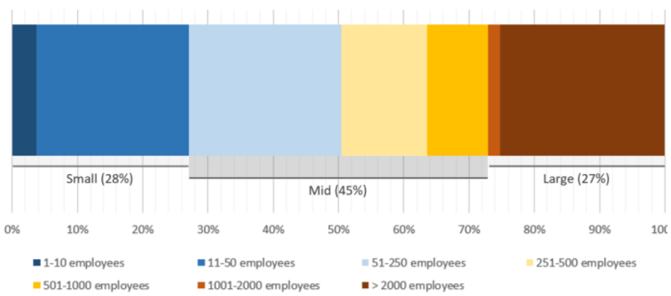

Figure 2: Organization's size

Most projects were described as agile (48%) followed by a hybrid process model (36%). Less common is the use of a traditional process (16%). Project teams (see Table 6) usually consisted of 10 to 20 people (31%), although 26% had fewer than five staff. In addition, a significant number (16%) indicated a team of more than 30 people. The median are teams with 5-9 people, indicating that participants tend to work in small development teams, but we have representatives from teams of all sizes.

Table 6: Team Size

Teams Size	#	%
< 5 people	28	26%
5 - 9 people	27	25%
10 - 20 people	33	31%
21 - 30 people	2	2%
> 30 people	17	16%

The most common system age was 2 to 5 years old (30%), closely followed by 1-2 years old (28%). We also had a significant number of represented systems of less than 1 year old (19%) and 5-10 years old (18%). Finally, the systems were typically between 10 KLOC and 1 million SLOC in size. But we also had good samples for smaller (<10 KLOC – 23%) and larger systems (>10 MLOC – 9%).

Thus, overall, the collected data seems to be a good representation of the Brazilian software industry diversity, reaching (i) several participants' roles and levels of experience, (ii) organizations of different sizes, and (iii) projects of different age, size, team size, and process models.

4.1 Familiarity with TD Concept (RQ1)

Initially, we asked (Q9) how familiar the participants were with the TD concept. We can see in Fig. 3 (left side) that for each ten participants, six are somewhat familiar with the concept. On the other hand, a large portion of the participants (40%) reported that they had never heard of it.

Next, we presented the TD definition adapted from McConnell [11] and asked (Q11) the participants how close this definition is to their understanding of the TD term. Most participants (80%) indicated that their understanding is close to or very close to the TD presented definition. This result seems to contradict the answers to the previous question, which indicated that 40% of the participants had never heard of it. However, the results suggest that, after being presented with a definition of the area, some of the participants who indicated that they had no previous exposure to the concept, actually had a personal intuition but were unsure about the terminology. The results for these two questions also indicate that although TD is a research topic close to the software industry, the use of the term still needs to be expanded.

We also asked (Q13) participants to provide an example TD item that occurred in their projects and, next, (Q15) how representative that example was. Most participants (42%) indicated that their example is a type of debt item that happens very often, closely followed by those who indicated that it was a kind of situation that occurs from time to time in the project (40%). Only 18% of participants indicated that the provided example is about a unique situation. These results indicate that debt problems upon which this study's results are based are real and recurrent in software projects.

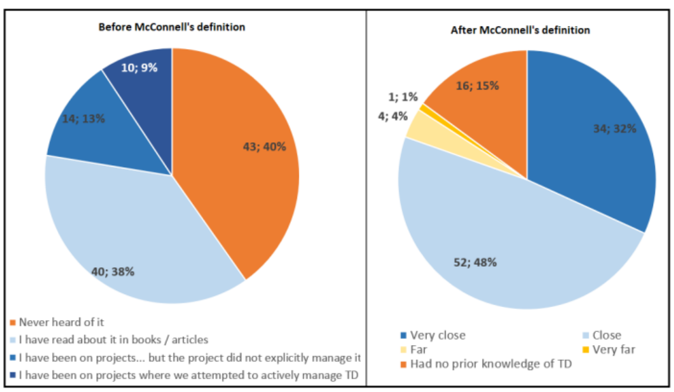

Figure 3: Participants' familiarity with TD concept

4.2 Causes of TD (RQ2)

The top 10 most cited TD causes, as informed by the 107 participants in Q16-18, are visualized together with the frequency in which they appear in the list of the causes most likely to lead to TD in Fig. 4.

We can observe that *deadline* is the most cited cause in general and as a most likely factor. This indicates that it is a factor that normally contributes to the occurrence of debt items. *Inappropriate planning, lack of knowledge, lack of a well-defined process, and non-adoption of good practices* are other causes cited by at least 20% of the participants. By analyzing Fig. 4, we can also observe that for some causes, even if they seem not to occur as often as others, they seem to be still more likely to lead to the occurrence of TD. For instance, lack of commitment is the tenth most frequently cited cause in the examples but it ranks in the sixth position among causes most likely to lead to TD in general. Therefore, it might be considered more important than,

Figure 4: Top 10 Technical debt causes cited

for example, *lack of knowledge*, which was mentioned more frequently in examples but less frequently in lists of causes most likely to lead to TD.

Results also indicate that technical reasons do not seem to be the most commonly remembered causes of TD. From the Top 10 cited causes and Top 10 most likely ones, the following causes are not directly related to technical issues: *deadline, inappropriate planning, lack of qualified professionals, team overload, no effective project management, lack of commitment, and lack of experience*. Thus, non-technical issues seem to have a significant role in the occurrence of TD items.

Another insight that we could observe in the data was that expertise and maturity of the team are decisive to the occurrence of TD, as demonstrated by the presence of the following cited causes: *lack of knowledge, lack of qualified professional, lack of experience, and lack of commitment*. Besides, planning and management issues (*deadline, inappropriate planning, and ineffective project management*) were cited often too.

4.3 Effects of TD (RQ3)

Fig. 5 shows the top 10 most cited TD effects (based on data from Q20 and Q21) together with the frequency in which they appear in the list of the effects that have a bigger impact (Q21). In this analysis, we do not present a specific figure for the top 10 effects that have a bigger impact because they were the same ranked as the top 10 most cited.

As we can observe, three effects stood out: *low quality, delivery delay,* and *low maintainability*, occupying the first three positions in the rank of the most cited effects and effects that have a bigger impact. This indicates that, besides being effects that normally impact the project, they are also the impacts that have major disruption to the project team. *Low quality* refers to any aspect that reduces the quality of an artifact (including errors and known defects that are not fixed). *Low maintainability* encompasses problems that occur during software maintenance activities, such as increased effort to fix bugs as well as limitation in system evolution. *Delivery delay* refers to the non-fulfillment of the deadlines agreed with the customer.

We can also observe in Fig. 5 that *rework and financial loss* are issues that commonly affect software projects in the presence of the debt and also are considered as having a bigger impact in the project. Finally, we could identify two effects related to relations among people: *team demotivation* and *stakeholders dissatisfaction*. This is a

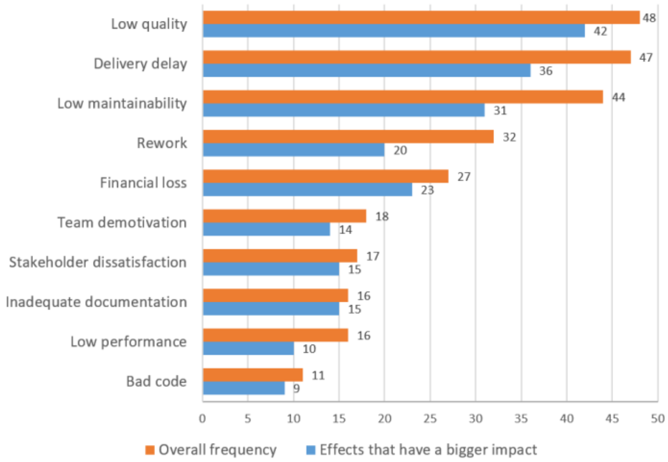

Figure 5: Top 10 Technical debt effects cited frequency

signal that the presence of TD can undermine the working environment.

Other cited effects are *inadequate documentation*, *low performance*, and *bad code*. *Inadequate documentation* encompasses issues related to software documentation such as, for example, incomplete, outdated, or nonexistent documentation. Low *performance* refers to issues in reaching performance requirements of the software (due to the degraded internal quality of the software). *Bad code* means the use of bad practices on coding activities (e.g., bad variables/methods names, over complex code).

5 DISCUSSION

5.1 Implications for practitioners and researchers

Development teams can use the lists of top 10 causes to understand factors that contribute to the occurrence of TD and, if necessary, work on preventive actions. For example, if a team recognizes such common causes as lack of knowledge or lack of experience as relevant to their own context, they could use avoidance of TD as further motivation to improve the workforce through hiring or training practices. The value of alleviating some of the non-technical causes, such as inappropriate planning or team overload, is enhanced by viewing these factors as causes of technical debt.

Furthermore, practitioners can have a clearer view of possible impactful consequences of TD in their projects by considering the lists of top 10 effects. This could aid decision making, by incorporating a wider range of potential consequences into decision models that try to capture the long-term cost of technical debt. Together, knowledge on causes and effects provide useful information to support decision makers on how to deal with TD. This set of information fills a gap in the technical literature.

For researchers, our results support future research by providing insights into software practitioners' perspectives on TD causes and effects. The presented top 10 lists allow researchers to guide their research in a problem-driven way. Finally, the global family of surveys allows not only researchers to reproduce the results and their interpretation, but also practitioners to evaluate their own TD situation against overall industrial trends.

5.2 Comparison to previous work

The initial results of InsighTD derived from its first execution in Brazil differs from current evidence on causes and effects of TD reported in related work in several ways. First, we could identify a comprehensive list of 79 causes and 66 effects of TD (the 10 most commonly cited and ranked as critical by practitioners were discussed in this paper). As we can observe in Fig. 6, this list considerably expands our understanding about reasons that lead development teams to incur TD on their projects and its implications for the teams.

Our results are based on answers from 107 practitioners from, at least, 80 different software organizations. This provides a much more representative set of results compared to related work that is based on small samples from a few organizations (except for [8], that has a large population size but concentrated on only three software organizations), as can be observed in Table 1.

Finally, regarding comprehensiveness of the results, one of our concerns in the design of InsighTD was to have a population that could express its opinion about the known types of debt. Thus, different from previous work that sometimes are concentrated on only one type of debt, or ask the opinion of participants about a predefined list of causes, this work intended to hear the voice of practitioners, trying to take a picture of their sources of pain (causes) and their pain itself (effects).

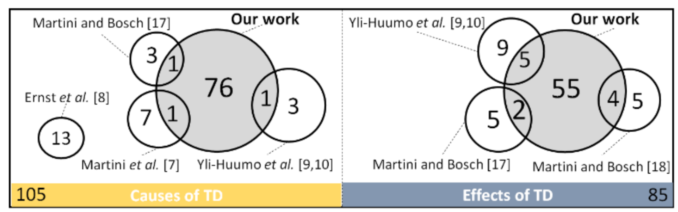

Figure 6: Overlapping of identified causes and effects among related works

6 THREATS TO VALIDITY

As in any empirical study, there are threats to validity in this work. We attempt to remove them when possible, and mitigate their effect when removal is not possible.

Construct validity: In this study, some social threats to construct validity can arise. These threats are concerned with issues related to behavior of the participants and the experimenters. Overall, participants may, based on the fact that they are part of a study, act differently than they do otherwise [19]. To help prevent hypothesis guessing and evaluation apprehension [19], in the invitation e-mail, we clearly explain the purpose of the study and ask the participants to answer questions based on their own experience. We also stated that the questionnaire is anonymous and that the collected data is analyzed without taking into consideration the participants' identities.

Concerning researcher expectations, in which experimenters can bias the results of a study both consciously and unconsciously based on what they expect from the experiment, we involved different researchers acting as internal and external reviewers of the questionnaire. Thus, this threat was also minimized.

Conclusion validity: This aspect is concerned with to what extent the data and the analysis are independent of the specific researchers [19]. In InsighTD, the major threat to conclusion validity arises from the coding process as coding is essentially a creative task. To mitigate this threat, we first conducted a pilot phase in the analysis. After agreeing on the first resulting codes (e.g., common understanding on the wording and the level of abstraction in the codes), the coding process was performed individually by two researchers. Then, they discussed the results until they reached consensus.

Internal validity: Maturation and instrumentation are two threats to internal validity that could affect this study. Instrumentation is the effect caused by the artifacts used for the study execution, in our case, the questionnaire. If it is badly designed, the study results are affected negatively [19]. To deal with this threat, the questionnaire was designed in a way that we have only direct questions and, thus, requiring as little interpretation as possible, avoiding a misunderstanding that would lead to meaningless answers. Besides, the questionnaire has passed through successive validation tasks (three internal and one external) and a pilot to detect any inconsistency or misunderstanding before running the survey.

Maturation indicates that the participants can react differently as time passes, for example, if the questionnaire is too long [19]. In our study, we tried to minimize this threat by avoiding a questionnaire that takes too much time to be answered. During the pilot study, we found that the mean time to complete the full questionnaire was about 20 minutes. Another sign that this threat was not raised was that all participants answered the questionnaire in full.

External validity: threats to external validity are conditions that limit our ability to generalize the results [19]. We reduce this threat by achieving diversity of participants who answered the survey. As discussed in Section 4, we had a population from different organization sizes, types of expertise, levels of experience and so on. However, although the population provides representative results on causes and effects of technical debt from the perspective of the Brazilian software industry, the results cannot yet be generalized. In search of more generalizable results, InsighTD is based on continuous replications of the questionnaire in different regions of the world. Currently, the survey is about to be replicated in United States.

Another threat that could affect this study is related to the lack of control over the participants invited to participate in the research. It could happen that only developers interested in the TD area answer the survey. This might bias the results towards a more positive view of TD knowledge. However, as presented in Section 4, about 40% of the participants initially indicated that they were not familiar with the concept of TD. Thus, we assume that this possible positive bias is not significant.

7 FINAL REMARKS

The contribution of this paper is twofold. First, we presented the design of a global family of surveys on causes and effects of TD on software projects. With InsighTD, we intend to reduce the problem of isolated investigations in TD that are not yet representative and, thus, build a continuous and generalizable empirical basis for understanding practical problems and challenges of TD. Second, we discussed the first results from the survey conducted in Brazil. In total, 107 practitioners answered the survey allowing us to investigate causes and effects of TD on software projects in the Brazilian software industry.

The next steps of this research include: (i) a deeper analysis to identify possible patterns of causes and effects in the collected data, (ii) investigation of how or if types of debt influence them, (iii) analysis of the questions of the survey not considered in this baseline report, and (iv) running other possible analyses that can be performed based on the characterization of participants. Also, the first replication of InsighTD is about to be performed in United States. The synthesis of both surveys will be further disseminated. Finally, other replications are planned by researchers already committed to InsighTD.

ACKNOWLEDGMENTS

This work was partially supported by the CNPq Universal 2014 grant 458261/2014-9, by the State of Bahia's SECTI-Fraunhofer-UFBa cooperation agreement 2012-1, and by the RESCUER project Grant: 490084/2013- 3.

REFERENCES

[1] Y. Guo and C. Seaman (2011), A portfolio approach to technical debt management, MTD 11: Proc. of the 2nd Workshop on Managing Technical Debt.

[2] P. Kruchten, R.L. Nord and I. Ozkaya, I. (2012). Technical Debt: From Metaphor to Theory and Practice. IEEE Software, Published by the IEEE Computer Society.

[3] N.S.R. Alves, T.S. Mendes, T.S., M.G. Mendonça, R.O. Spínola, F. Shull and C. Seaman (2016). Identification and Management of Technical Debt: A Systematic Mapping Study. Information and Software Technology, 70, 100 – 121, 2016. https://doi.org/10.1016/j.infsof.2015.10.008

[4] C. Seaman. Qualitative methods in empirical studies of software engineering. IEEE Transactions on Software Engineering, 25(4):557-572, 1999

[5] A. Strauss and J. M. Corbin. Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. Sage Publications, 1998.

[6] Y. Guo, R. O. Spínola, C. Seaman (2014) Exploring the costs of technical debt management – a case study, Empirical Software Engineering, v. 1, p. 159-182. DOI=http://dx.doi.org/10.1007/s10664-014-9351-7

[7] A. Martini, J. Bosch, and M. Chaudron. 2014. Architecture Technical Debt: Understanding Causes and a Qualitative Model. In Proceedings of the 2014 40th Euromicro Conference on Software Engineering and Advanced Applications (SEAA '14). IEEE Computer Society, Washington, DC, USA, 85-92.

[8] N. A. Ernst, S. Bellomo, I. Ozkaya, R. L. Nord, and I. Gorton (2015). Measure it? Manage it? Ignore it? software practitioners and technical debt. In Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2015). ACM, New York, NY, USA, 50-60.

[9] 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. In: PROFES 2014. LNCS, vol. 8892, pp. 93–107. Springer, Heidelberg (2014). doi:10. 1007/978-3-319-13835-0_7

[10] J. Yli-Huumo, A. Maglyas, and K. Smolander. The benefits and consequences of workarounds in software development projects. In: Fernandes, J.M., Machado, R.J., Wnuk, K. (eds.) ICSOB. LNBIP, vol. 210, pp. 1–16. Springer, Heidelberg (2015).

[11] S. McConnell, "Technical debt," 10x Software Development Blog (2007). Construx Conversations. URL= http://blogs.construx. com/blogs/stevemcc/archive/2007/11/01/technical-debt-2.aspx, 2007.

[12] N. Rios, M.G. Mendonça, and R.O. Spínola, 2018, A tertiary study on technical debt: Types, management strategies, research trends, and base information for practitioners, Information and Software Technology, Volume 102, Pages 117-145, ISSN 0950-5849, https://doi.org/10.1016/j.infsof.2018.05.010.

[13] Agilealliance.org, 'Manifesto for Agile Software Development'. [Online]. Avail.: https://www.agilealliance.org/agile101/the-agile-manifesto/. [29-Mar-2018]

[14] Agilealliance.org, 'What is Hybrid Agile, Anyway?'. [Online]. Available: https://www.agilealliance.org/what-is-hybrid-agile-anyway/. [29-Mar-2018]

[15] R. S. Pressman and B. R. Maxim. 2014. Software Engineering: A Practitioner's Approach (8th ed.). McGraw-Hill Higher Education

[16] J. Yli-Huumo, A. Maglyas, and K. Smolander. 2016. How do software development teams manage technical debt? - An empirical study. Journal of System and Software. 120, C (October 2016), 195-218

[17] A. Martini and J. Bosch. The Danger of Architectural Technical Debt: Contagious Debt and Vicious Circles. In Proc. of the 2015 12th Working IEEE/IFIP Conf. on Software Architecture. IEEE Computer Society, Washington, DC, USA, 1-10.

[18] A. Martini and J. Bosch. Towards prioritizing Architecture Technical Debt: information needs of architects and product owners. In: 41st Euromicro Conf. on Soft. Eng. and Advanced Applications (SEAA), 2015. IEEE, 2015. p. 422-429.

[19] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and Anders Wesslén. 2012. Experimentation in Software Engineering: An Introduction. Springer.