

Investigating the Effects of Agile Practices and Processes on Technical Debt - The Viewpoint of the Brazilian Software Industry

Viviane Coelho Caires
PPGCOMP, Salvador University
Federal Institute of Bahia-IFBA
Jequié/Vitória da Conquista, Brazil
viviane.caires@ifba.edu.br

Nicolli Rios
Department of Computer Science,
Federal University of Bahia
Salvador, Brazil
nicollirios@gmail.com

Johannes Holvitie
TUCS - Turku Centre for Computer
Science & University of Turku
Finland
jjholv@utu.fi

Ville Leppänen
TUCS - Turku Centre for Computer
Science & University of Turku
Finland
ville.leppanen@utu.fi

Manoel G. de Mendonça Neto
Department of Computer Science,
Federal University of Bahia
Fraunhofer Project Center @ UFBA
Salvador, Brazil
manoel.mendonca@ufba.br

Rodrigo Oliveira Spínola
PPGCOMP, Salvador University
Fraunhofer Project Center at UFBA
Salvador, Brazil
rodrigo.spinola@unifacs.br

Abstract—The current scenario of software development is characterized by a wide adoption of agile methodologies, which define processes and practices that address a range of problems faced by development teams. However, there is still little information on how these methodologies deal with technical debt(TD). This work presents the results of a replicated survey(originally executed in Finland) whose goal was to investigate which agile practices and processes are sensitive to TD. Despite this replication allows different types of analysis, the focus of this paper will be on the analysis of the effects of the agile practices and processes on TD from the perspective of the Brazilian software industry, where the study was replicated. At total, 62 practitioners from different organizations answered the questionnaire. The results indicated that participants already had a good knowledge about TD, instances of TD reside in the software implementation and are caused due to deficiencies in its architecture, the size of a debt item is proportional to its impact on the project, and, refactoring and iteration have the most positive effect on TD. This replication also contributes to the investigated topic through the accumulation of evidence about the findings, thereby increasing the level of confidence in results.

Keywords—*Technical debt; agile methodology; survey; replicated study.*

I. INTRODUCTION

Technical debt (TD) represents the effects of immature artefacts that bring short-term benefits in terms of increased productivity and lower costs, but which may need to be adjusted later with interest during software development [1, 2, 3]. TD is usually incurred when development teams have to choose between to evolve the system considering quality standards or to put it to run in the shortest possible time, using minimum resources. As TD is incurred in a project, the effort required to eliminate it is cumulative and its payment tends to become more complex. Different types of debt may occur during the phases of a software development process and the used methodology can affect their presence [4]. An inadequate

management of TD can bring significant losses to a software project [7].

The current scenario of software development is characterized by a wide adoption of agile methodologies, which define processes and introduce practices that address a range of problems currently faced by development teams [5]. However, there is still little information on how these methodologies accommodate the concept of TD. To shed some light in this discussion, Holvitie *et al.* [9] conducted a survey with practitioners from Finland on how TD issues relates to agile software processes and practices. They investigated participants' level of knowledge on TD, how TD manifests itself in their projects, and what processes and practices of agile development are sensitive to it. In general, the study pointed out that the processes and practices that are closest to the implementation and maintenance activities are perceived as having the most positive effects on the control of TD. In addition, the authors also identified that TD items usually come from problems in the software architecture.

Although the surveys' results are valid, the work of Holvitie *et al.* [9] is also limited by some issues. The main one is that the data were collected from development companies based in Finland and, therefore, the results may reflect only the local scenario. To deal with this, an international consortium involving researchers from Finland, Brazil and New Zealand worked together to replicate the survey in their respective countries. The goal of this set of replications was investigate whether the findings of the Holvitie's study are reproducible. Results from the whole gathered data were reported in [10]. However, as the results presented in [10] did not consider the specificities of each involved country (the whole dataset was analyzed as an only instance), an in-depth analysis of the results of each individual replication is still missing. This kind of analysis can reveal hidden details that could not be perceived in a generic look at the data. More specifically, it can reveal how software practitioners perceive impacts that agile practices

and processes have in TD considering the local software industry reality that usually differs from other places when we consider variables like the size of organizations and development teams, and the size and duration of software projects.

This work presents the results of the replication¹ of the study of Holvitie *et al.* [9] in Brazil, a country located in another continent and with a culture different from Finland. Despite the fact of this replication allows different types of analysis (for example, comparison between the results from each country), the focus of this work is to discuss the results of the replication in Brazil. Thus, we will present an analysis on the effects of the agile software development practices and processes on TD from the perspective of the Brazilian software industry. We will discuss the answers to the following research questions: **RQ1** - *What is the level of knowledge of respondents about TD?*; **RQ2** - *Which agile software development practices and processes are sensitive to TD?*; and **RQ3** - *How does TD manifest itself in the participants' work?*

To replicate the survey, we used a web-based questionnaire that was answered by 62 practitioners (mostly characterized by professionals with more than 6 years of experience) from 62 different software organizations. In general, the results indicated that the participants already had a good knowledge about the concept of TD, but some of them are still not familiar with the term. In another finding, we could observe that many instances of TD reside in the software implementation and are caused due to deficiencies in its architecture. We also identified that the size of a debt item is proportional to its impact on the project. Finally, considering all analyzed agile software development practices and processes, most of respondents indicated that refactoring (practice) and iteration (process) have the most positive effect on TD.

In addition to this introduction, this paper has five more sections. In Section II, the replication of the survey in Brazil is described. The effects of the agile software development practices and processes in TD are presented in Section III. Section IV discusses the obtained results. Next, limitations of the study are presented in Section V. Finally, Section VI presents some final remarks.

II. SURVEY REPLICATION - BRAZIL

A. Survey

The goal of the research performed by Holvitie *et al.* [9] was to investigate which agile software development practices and processes are sensitive to TD. They conducted a survey, structured in three groups of questions, considering a population of practitioners.

The first group of questions aims to establish the level of knowledge of the respondents about software development and how they perceive TD in their projects. For this, the research questions were defined as, **for an individual**: (**RQ1.1**) does work experience, (**RQ1.2**) do used agile development practices, or (**RQ1.3**) do associated project responsibilities correlate with

what the respondent perceives his/hers assumed or actual TD knowledge to be?; (**RQ1.4**) in which mediums has he/she seen or heard the term TD be used?; (**RQ1.5**) in which situations has he/she or his colleagues applied the concept of TD?, and; (**RQ1.6**) in which situations does he perceive the use of the TD concept as helpful? During this first stage, the authors also present the McConnell's definition of TD [7], ensuring that all participants know the term.

In the second stage, there is a set of questions about which agile development practices and processes are used by respondents in their projects and how they realize that their use affects TD. We established that the XP practices together with Scrum processes cover the components of agile software development well in addition to being highly popular [11, 12]. Questions of this stage intend to answer the following research questions: **are there certain agile software development practices or processes for which** (**RQ2.1**) their effect on technical debt is seen to be significantly positive, neutral or negative?; (**RQ2.2**) it is seen that they (do not) cover the team's or the project's development management needs?, and; (**RQ2.3**) it is seen that they (are not) able to cover TD issues that require management?

In the third stage of the survey, participants are asked to cite particular instances of TD and, from that concrete instance, answer the following research questions: **for a concrete instance of technical debt**, (**RQ3.1**) in which phase of the software development it was observed?; (**RQ3.2**) what are the causes for its emergence?; (**RQ3.3**) is it legacy?; (**RQ3.4**) is its size dynamic?, and; (**RQ3.5**) does its effects correlate with its size?

In total, the questionnaire has 37 questions (35 objective and 2 subjective) and collects the following information: (i) participants' knowledge on software development; (ii) organizational details (such as participants' role in the project, number of projects developed by the company, number of people involved in a given project); (iii) agile development processes and practices that are applied; (iv) interviewee's knowledge on TD; (v) perception of the development phases affected by TD; and (vi) an example of an artifact affected by TD, the size of that debt item and its perceptible effects. The survey, available at <http://soft.utu.fi/tds16/questionnaire.pdf>, was developed as a web-based form in order to increase the response rate and minimize data manipulation errors. Google Forms platform was used for building, distributing, and collecting survey data.

B. Survey Brazil

When we decided to replicate the survey in Brazil, it was already designed and all the instruments were available. Therefore, in this section we focus on the details of how we planned and operated the replication in Brazil. Further information on the design of the survey can be found in [9].

To plan the survey replication in Brazil, we held a couple of discussions with the general organizers. During the discussions, the online questionnaire was presented and some general guidelines for conducting the survey were provided. Thus, the configuration of the environment was performed and then the participants were invited by e-mail to contribute with the

¹ Replication based on previous insights is widely recommended in the experimental paradigm [13]

research. Participants were selected through software associations or local industry contacts. In this process, we tried to reach practitioners spread out in different regions in Brazil.

In total, 62 professionals from different software development organizations answered the survey. Regarding the size of the organizations in terms of number of employs, 44% of the respondents work on organizations with over 250 employs. A significant number (30%) of answers were also obtained from participants of organizations that have between 10 and 50 employs. 10% of the participants work on small companies with less than 10 employs. Finally, 16% of the respondents indicated that work on companies that have between 51 and 250 employs. The development teams in which participants are involved in are mainly characterized as small teams (42%, 2-5 members). 25% of the respondents work on teams that have between 6-10 members. We also had answers from teams with over 20 members (10%). The other participants are part of very small or middle size teams.

The length of projects in which participants are working on has the following distribution: 1-3 months (23%), 4-6 months (31%), and over 6 months (38%). Regarding development iteration length, the answers are distributed as follows: one week or less (18%), 2-3 weeks (26%), 1 month (10%), 2 months (5%), over 2 months (10%), no iteration (20%). Finally, concerning respondent level of experience, approximately 15% of the respondents have less than 3 years of software development experience, slightly more than 20% have between 3 and 6 years, and 65% have more than 6 years. The average time to complete the questionnaire was 15 minutes.

To ensure a standardized data analysis in relation to the work of Holvitie *et al.* [9] and make possible a future comparison between them, we forwarded the responses to the general organizers, which applied the same analyzes carried out in the study performed in Finland. Then, they returned the results and we could interpret them.

III. RESULTS

In this section, we discuss the survey results concerning (i) what is the level of knowledge on TD of the participants, (ii) what agile software development practices and processes are perceived as sensitive to TD, and (iii) how TD manifests itself in their work.

A. RQ1 - What is the level of knowledge of respondents about TD?

Research questions grouped by RQ1 are focused on participant's perception on the concept of TD. For this, participants are initially asked about how they perceive their knowledge on TD, followed by a request for them to (optionally) describe their definition of the term. These answers were classified according to respondents' work experience (RQ1.1), applied software development techniques (RQ1.2) and assumed roles (RQ1.3). There was no significant difference between the distributions of these variables. Thus, the most general one is presented here. From Figure 1, it is observed that 32% of the respondents considered having a good or very good definition of TD, however, almost 50% of them

indicated that they did not know the term or had a poor definition of it.

Next, the McConnell's definition of TD [7] was presented and, then, the respondent was asked to indicate how close to this concept was his initial understanding. The results are represented in Figure 1 and indicate that about 70% of respondents reported that their definition were close to or very close to the definition extracted from the technical literature. Besides, slightly more than 20% reaffirmed not knowing the term or having a poor definition of it. These data indicate that participants were initially reticent about their understanding on concept of TD, but that most of them (80%) really already knew it. Other surveys performed in the area have pointed out this same behavior [6][9].

Complementing this analysis, Figure 2 presents the relationship between the experience of survey participants, their previous knowledge on TD and their knowledge after the definition be presented in the questionnaire. We can see that for interviewees with less than 3 years of experience, 5% had a good or very good definition for TD, 8% had a poor or very poor definition, and 2% reported not knowing the term. After reading the definition presented in the survey, the percentages passed to 11%, 2% and 2%, respectively. For participants who had experience between 3 and 6 years, 3% indicated having a good or very good definition for TD, 16% a poor or very poor definition and 2% indicated not knowing the term. After reading the definition, the percentages passed to 10%, 2% and 10%, respectively. Finally, for the most experienced participants (more than 6 years of experience), 25% reported having a good or very good definition for TD, 29% had a poor or very poor definition, and 11% reported not knowing the term. This percentage changed to 49%, 10% and 6%, respectively, after reading the TD concept presented.

Then, the respondents were asked where they had either

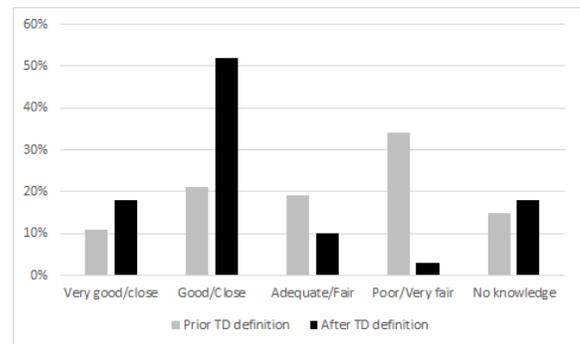


Figure 1. Distribution for perceived TD knowledge

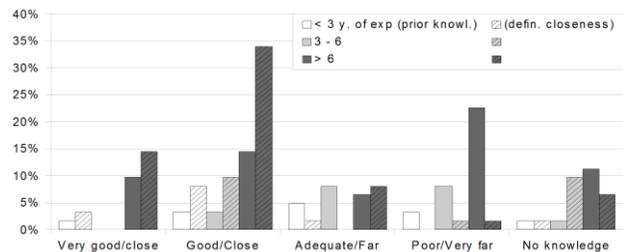


Figure 2. Relationship between interviewees' experience and knowledge on TD concept

seen or heard the term TD used (RQ1.4). The questionnaire provided seven initial options that can be observed in Figure 3. We can see that more than 50% have seen the term in the technical literature. Surprisingly, about 40% of the respondents reported the term has been used in work meetings. It is also important to mention that over 15% of respondents never had heard the term before.

Finally, closing the analysis of RQ1, a mapping of common

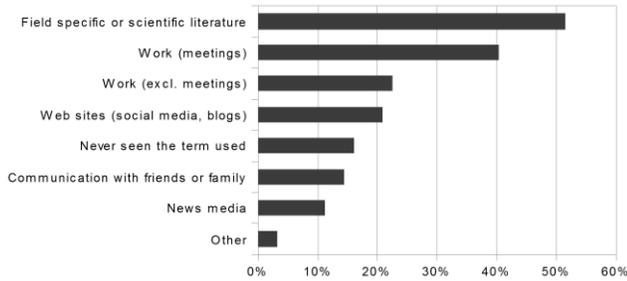


Figure 3. Technical debt usage in different mediums

decision situations in development is shown in Figure 4. We asked participants whether, for each situation, he/she or a co-worker had already applied the concept of TD (RQ1.5) and whether the use of the concept would have been useful (RQ1.6). The data show that more than half of the respondents realized the utility of using the concept of TD in all situations and only 5% reported that its use would not bring gains.

Still on Figure 4, 27% of respondents reported that they have already applied the concept of TD in unforeseen situations, almost 20% in decisions about development infrastructure, about 20% in integrated resources, and 35% in conduction of software development. From the perspective of a co-worker, 27% reported that a colleague had already used the term TD in unforeseen situations, 18% in issues involving development infrastructure, 10% in integrated resources, and almost 30% in the conduction of software development. It is worth mentioning that more than 50% of respondents never used the TD concept in decision-making in any of the situations.

Finding 1: The concept of TD is already known by a large part of the population represented in this study. On the other hand, practitioners are still assimilating the concept.

Finding 2: The usefulness of using the TD concept in development activities is recognized.

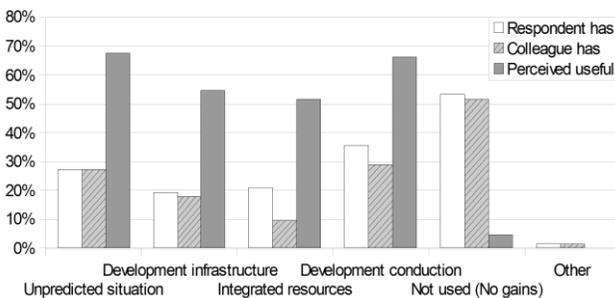


Figure 4. Respondent's application and perceived usefulness of applying the TD concept

B. RQ2 - Which agile software development practices and processes are sensitive to TD?

The RQ2 group of questions is focused on investigating the effects of agile software development practices and processes on TD. For this, initially the questionnaire presented a list of 11 agile development practices [11] and, for each of them, the interviewee should indicate how positively/negatively it could impact the TD in the project (RQ2.1). The results presented in Figure 5 demonstrate that practices used during implementation phase (simple design, TDD, coding standards, refactoring, continuous integration, and pair programming) are considered by more than half of the interviewees as having a positive or very positive effect on TD. More specifically, refactoring was indicated as the practice that has the most positive effect.

Afterwards, we asked participants about the effect of agile development processes on TD (RQ2.1). For this, a list of six processes [12] was considered. As we can see in Figure 6, all processes (iteration planning meetings, iterations, iteration backlog, iterations reviews/retrospectives, daily meetings, and product backlog) were considered to have very positive or positive effects on TD. The iteration process was considered the most positive among them.

We also asked if the combination of agile techniques that participants used were adequate for the team's or the project's management needs (RQ2.2) and if the techniques were able to cover all aspects that require management (RQ2.3). For singular practices, processes and their adoption rates, not a single combination could be identified for which the difference in their management or cover characteristics was statistically significant.

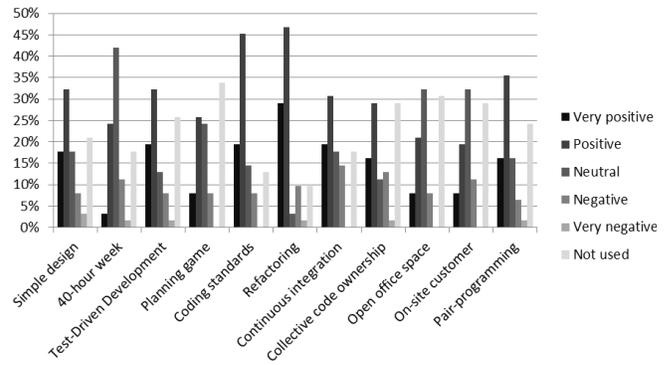


Figure 5. Perceived effect of agile software development practices on TD

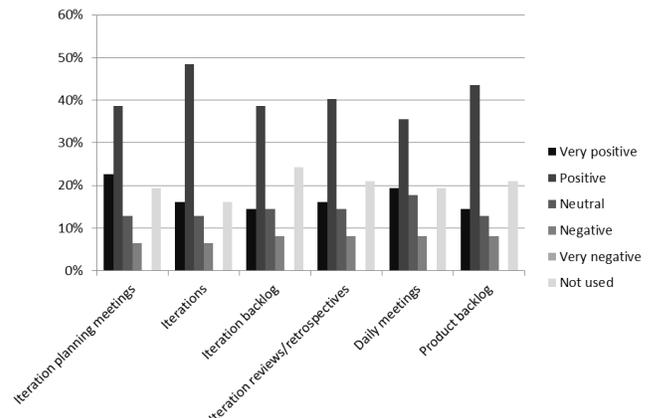


Figure 6. Perceived effect of agile development processes on TD

Finding 1: In general, agile practices have a positive effect on the TD. Among the analyzed practices, refactoring was considered to have the most positive effect.

Finding 2: In general, agile processes have a positive effect on the TD. Among the considered processes, iteration was indicated as having the most positive effect.

RQ3 - How does TD manifest in the participants' work?

The group of questions RQ3 is focused on the analysis of situations that TD affected the progress of projects in which the participants were involved. When asked about in which phase of the software development the TD was observed (RQ3.1), as we can see in Figure 7, 77% of respondents stated that the implementation phase is the most affected, followed by design phase. Although the testing phase was reported as the least affected, its percentage is still relevant.

We also investigated the causes that led to the occurrence of debt (RQ3.2). To do this, from a previously defined list of causes [2], the participant should indicate which of them he/she considered pertinent. In Figure 8, we can see that the causes most often indicated by participants were inadequate architecture and inadequate structure, followed by violation of best practices or style guides, and inadequate testing and documentation. This result is aligned with findings reported by Ernst *et al.* [6] that also pointed to problems in architecture as the main source of TD in software projects.

In addition to this question, when asked about the source of TD instances (RQ3.3), most participants (50%) stated that TD instances came from the legacy from an earlier team/individual who previously worked on the same project/product. 18% of the participants indicated that their source is in the legacy from an unrelated project/product of the organization, and 14% stated that the source is in the legacy from outside the organization. Only 18% of participants answered that the source is not from legacy activities.

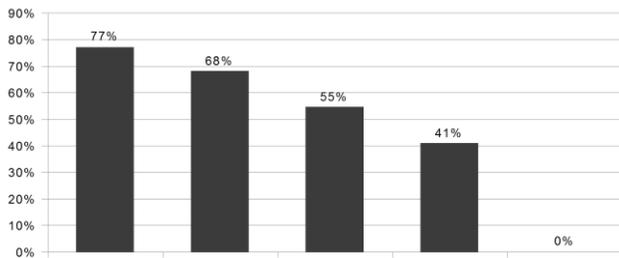


Figure 7. Distribution of TD by project phases

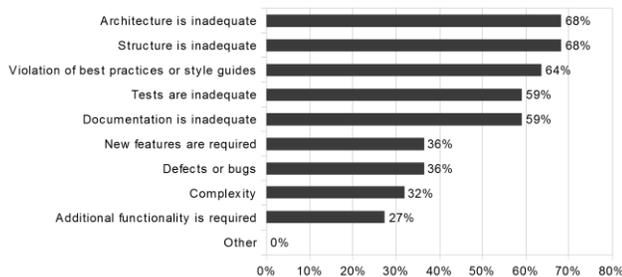


Figure 8. Indicated causes for concrete instances of TD

Then, when asked about the relationship between the continued development of a component and the size of the debt located in that component (RQ3.4), most of participants (82%) reported that the continued development would contribute to the increase in the size of the debt, while only 18% reported that this would lead to a reduction in the size of the debt. None of the respondents indicated that there would be a large decrease or no change.

Finally, when asked about the correlation between the size of a TD item and the effects that it causes in software development (RQ3.5), about 70% of the respondents answered that the size is directly proportional to the magnitude of the effects, 5% stated that it is not proportional, and another 5% answered that it is inversely proportional. Other 23% reported that the size is somehow proportional to effects magnitude. Thus, for most respondents, the larger is the size of a debt item, the greater is the effect it brings to the project.

Finding 1: The implementation phase is, usually, the most affected by TD.

Finding 2: Inadequate software architecture and internal structure are the main causes of TD.

Finding 3: Debt items tend to come from the legacy of a team/individual who previously worked on the same project/product.

Finding 4: Continued development in a software item contributes to the increase in TD's size on that item.

Finding 5: The larger the size of a TD item is, the greater is the effect it brings to the project.

IV. DISCUSSION

This work presented the results of a replicated survey in Brazil. For RQ1, we found that the concept of TD is already known by a good part of the population represented in this study. Regarding RQ2, we observed that, in general, agile software development practices and processes have a positive effect on TD. In this item we highlight the refactoring practice and the iteration process, which were considered as having the most positive effect. Finally, for RQ3, the data indicated that the implementation phase is the most affected by debt items, and problems associated with the architecture and internal structure of the software are the main causes of TD. These results justify, at a certain extension, the fact that agile practices that have a more positive effect on TD are directly related to coding activities.

Another result from RQ3 indicated that continued development in a software item contributes to the increase in TD in that item. This is an interesting result because if, on the one hand, continuous work on an item opens opportunities for improvements in its internal structure (that can lead to payment of debt items), on the other hand, if we do not explicitly manage TD, these opportunities can be lost and, as consequence, the debt size can reach higher levels. Finally, participants also reported that there is proportionality between debt size and the effects it brings to the project. These two

results reinforce the importance of making explicit the management of the TD items.

A. Relation to previous work

The results of this replication indicated that the population has different characteristics from the original study [9]: (i) Finland sees a majority in the smaller organization size categories and Brazil is in the middle ground with highs in medium and large categories; (ii) Finland had average iteration length of two to three weeks whereas Brazil is more evenly distributed (from 1 week to more than 2 months), and (iii) projects in Brazil tend to be longer and Finland's shorter. Thus, on the perspective of the population characterization, we could say that this replication contributes to the original study by expanding the sample from the organization spectrum. Besides, it also indicates that the obtained results reflect particularities of the Brazilian local scenario of agile software development. Concerning participants' level of experience, country-wise deviation is almost non-existent.

Regarding results for research questions, despite in general both executions pointed out to the same direction, we also could detect particularities. For example, respondents from both countries answered that common agile practices and processes are sensitive to technical debt. However, while in Brazil refactoring and iteration are considered as having the most positive effects on TD, in Finland, participants indicated coding standards and iteration reviews/retrospectives. A more detailed analysis of differences and similarities between the results obtained with the execution of the survey in Finland and its replication in Brazil is out of the scope of this paper, being part of the next steps of this research.

V. STUDY LIMITATIONS

Some limitations apply to this study. One of them is related to the cultural influence of the region where the survey was performed. Although the questionnaire was answered by participants from different companies of different sizes and based in different regions from Brazil, yet the "Brazilian way" of developing software may have influenced the responses. A detailed analysis considering this aspect is outside the scope of this paper, but it is a future work that will be carried out by the authors considering the data obtained with the execution of the survey in Finland and its replication in Brazil.

A second limitation that affects 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 participate of the study. This might bias the results towards a more positive view of technical debt knowledge. However, about 50% of the respondents initially indicated that they were not familiar with the concept and thus we assume that this positive bias is not significant.

Finally, although the number of responses (62) can be considered good, yet the data cannot be generalized to represent practitioners from Brazilian software industry. Still, they provide valuable indicators on the research questions raised. Their analysis together with the data obtained from the original execution of the study in Finland will allow a greater level of confidence in the results.

VI. FINAL REMARKS

This work is aligned with a growing concern of the software engineering community: the replication of empirical studies. It contributes to the generation of knowledge in a given topic through the accumulation of evidence about the findings, thereby increasing the level of confidence in results [13].

Specifically, this replicated study investigated the perception of practitioners on TD concept, the effects of agile software development practices and processes on it, and how TD manifests itself in practice in the Brazilian software industry. The reached results, described in details on Sections III and IV, contribute to the improvement of the body of knowledge that has been built around the Technical Debt Landscape [3] [8]. The next steps of this research include a country level comparison of the obtained results.

ACKNOWLEDGMENT

This work was partially supported by the CNPq Universal 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] C. Seaman & Y. Guo (2011), Measuring and Monitoring Technical Debt, *Advances in Computers* 82, 25-46.
- [2] P. Kruchten; R. Nord & I. Ozkaya (2012), Technical Debt: From Metaphor to Theory and Practice, *Software*, IEEE 29(6), 18-21.
- [3] C. Izurieta; A. Vetro; N. Zazworka; Y. Cai; C. Seaman & F. Shull (2012), Organizing the technical debt landscape, in *Managing Technical Debt (MTD)*, 2012 Third International Workshop on, pp. 23-26
- [4] N.S.R. Alves, T.S. Mendes, M.G. Mendonça, R.O. Spínola, and C. Seaman, Identification and management of technical debt: A systematic mapping study, *Information and Software Technology*, Volume 70, February 2016, Pages 100-121, ISSN 0950-5849.
- [5] R. C. Martin, *Agile software development: principles, patterns, and practices*. Prentice Hall PTR, 2003.
- [6] 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 *Proc. of the 2015 10th Joint Meeting on Foundations of Software Engineering*. ACM, New York, NY, USA, 50-60.
- [7] S. McConnel, "Managing Technical Debt," *Construx Software*, Version 1. Available from: <http://www.construx.com>, 2008
- [8] N.S.R. Alves, R.S. Araújo, R.O. Spínola. A Collaborative Computational Infrastructure for Supporting Technical Debt Knowledge Sharing and Evolution. In: *Americas Conference on Information Systems*, 2015, Puerto Rico.
- [9] J. Holvitie; V. Leppanen & S. Hyrnsalmi (2014), Technical Debt and the Effect of Agile Software Development Practices on It - An Industry Practitioner Survey, in *MTD 2014*, pp. 35-42.
- [10] J. Holvitie, S.A. Licorish, R.O. Spínola, S. Hyrnsalmi, S.G. MacDonell, T.S. Mendes, J. Buchan, and V. Leppänen. Technical debt and agile software development practices and processes: An industry practitioner survey. *Information and Software Technology*, 2017, ISSN 0950-5849.
- [11] N. Kurapati, V. S. C. Manyam, K. Petersen, Agile software development practice adoption survey, in: *Agile processes in software engineering and extreme programming*, Springer, 2012, pp. 16{30.
- [12] D. West, T. Grant, Agile development: Mainstream adoption has changed agility 2 (41).
- [13] F. Shull, J.C. Carver, S. Vegas, and N. Juristo. 2008. The role of replications in Empirical Software Engineering. *Empirical Software Engineering*. 13, 2 (April 2008), 211-218.