Investigating Technical Debt Folklore: A Replicated Survey

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

José Amâncio Macedo Santos
State University of Feira de Santana
Feira de Santana, Brazil
zeamancio@uefs.br

Manoel Mendonça
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
State University of Bahia
Salvador, Brazil
rodrigo.spinola@unifacs.br

Abstract—**[Context] The software engineering community considers the technical debt (TD) concept intuitive, because it facilitates discussion among team members about problems that can impact the software development. Personal opinions and experiences related to the concept have been published in blogs and other channels without any evaluation, originating the TD Folklore. [Goal] This work aims to investigate TD Folklore statements classifying them by agreement and consensus. Besides, we also investigated if software development experience affects the perception of developers. [Method] We replicated a survey to evaluate TD Folklore statements. In the replication, we increased the number of respondents and added a new research question to analyze the difference of opinions between participants with and without software experience. [Results] At total, the survey was answered by 107 respondents. The list of TD Folklore was reorganized by the ranking of agreement and consensus indicated by participants. We also identified that professional experience does not change the participants´ perception on the concept of TD for the most cases. [Conclusion] We believe that TD Folklore can help researchers and practitioners identify gaps for new research efforts.**

Keywords-Technical debt; technical debt folklore; survey

I. INTRODUCTION

Ward Cunningham cited technical debt (TD) for the first time in 1992 as: "shipping first time code is like going into debt. A little debt speeds development so long as it is paid back promptly with a rewrite. The danger occurs when the debt is not repaid. Every minute spent on not-quite-right code counts as interest on that debt." [5]. Since then, the concept, that originally had its scope limited to source code issues, has been expanded and considered in different stages of a software development project [1][2].

Currently, it is common to see subjective opinions, personal points of view and catch phrases about TD in blogs and websites. All this attention-grabbing information has raised concerns, since it was not evaluated before it was published and reflects only the opinions and experience of the authors. This scenario characterized by different and contradictory opinions, but without any evaluation, could led to the emergence of TD Folklore [12]. The term folklore corresponds to traditional stories, beliefs and customs of a group of people. TD Folklore needs to be investigated, because it mays contain valuable information about experiences that could contribute significantly to the study of the area. Thus, it can help researchers formulate theories and hypotheses, and identify gaps to direct new research efforts.

In this context, Spínola *et al.* [12] conducted searches on the Internet looking for TD folklore statements. The search was performed on online websites, blogs, and published papers. As result, the authors selected a list of 14 potential TD folklore statements. After, the authors performed a survey with the purpose of answering the following research questions:

(RQ1) Agreement: With which folklore statements did participants agree or disagree?

(RQ2) Consensus: How strong is the consensus on each of the folklore statements?

The rationale behind these questions was that if any folklore is either widely agreed to or disagreed with by a large group of people, then those propositions are more likely to be good candidates for future research. On the contrary, mixed responses can indicate that a TD Folklore item is not commonly believed, depends on many factors, or that the statement itself is not yet formulated as precisely as needed.

The results initially presented gave us interesting insights on the subject. From then, we extend the work of Spínola *et al.* [12] by replicating their study[1]. In this paper, we present this replication, which has two main objectives: i) to mitigate

[1] Replication based on previous insights is widely recommended in the experimental paradigm [3][13][14].

http://dx.doi.org/10.18293/SEKE2018-014

limitations of the previous study and; ii) to expand the knowledge on the topic. We address these points as follows. First, the main limitation of the previous study presented by Spínola *et al.* [12] is related to the number of participants. In this replication, we expand the population of the study from 37 to 107 (70 new participants). Second, the insights of the previous survey indicated that we need to investigate how the developers' experience impacts on TD Folklore statements. Then, in this replication, we also revisited the original research questions including the following new question:

(RQ3) Behavior: Do participants with and without software development experience have the same perception on TD folklore statements?

By answering RQ3, we intend to collect evidence that could provide some support for a known claim in the TD area: one of the advantages of the TD concept is that it has a common understanding in the software development community. We want to analyze this claim by investigating if the lack of industry experience affects the perception of software engineers about TD folklore items.

As a result of this replication, we highlight that the agreement and consensus analysis (RQ1 and RQ2) reinforce previous findings reported by Spínola *et al.* [12]. We also found that the participants' experience (RQ3) does not affect their agreement with the TD folklore statements, for the most cases. This builds evidence on the idea that TD concept has a common understanding among developers and represents a simple metaphor to discuss software development problems.

The contribution of this paper is twofold. First, we reevaluated and reorganized the TD Folklore list by rank of agreement and consensus. We also complemented this analysis by investigating if experience on software development activities influences the perception of participants about each TD folklore statement. Second, this replicated study will help us in understanding what participants have said about TD and what folklore seems to make sense and constitute good candidates for more detailed investigation.

In addition to this introduction, this paper has other five sections. Section 2 presents a background on the area. Then, in Section 3, the TD Folklore's survey will be presented. Section 4 discusses the results of this replication. Next, Section 5 presents some threats to validity of this study. Finally, Section 6 presents the final remarks of this work.

II. BACKGROUND

Different surveys have been conducted in the TD area. Klinger *et al.* [9] interviewed four experienced software architects to investigate how decisions on incurring debt are taken within a company and what is the extent of the consequences of those decisions. They concluded that often the decision to incur debt is not direct action of the architects, but a consequence of activities carried out by people that do not perform technical activities in the project.

In another study, Lim *et al.* [10] characterized how software professionals perceive and understand the context in which TD occurs. After interviewing 37 professionals, they concluded that to deal with the balance related to TD, professionals must make this debt explicit, communicate their costs and benefits to all stakeholders and manage it making its presence healthy for the project.

Snipes *et al.* [11] conducted an interview with the change management committee regarding the defect debt management. As a result, they indicated that the highest cost of this type of debt was related to their identification and validation activities (cost of testing). In addition, the authors also identified that there are six major components which affect decisions about incurring/paying a debt item: severity, existence of an alternative solution, urgency of the correction, effort to implement the correction, risk of the proposed correction, and the extent of the required test.

In another study, Codabux and Williams [4] conducted a survey to identify best practices with respect to TD management. They analyzed 28 teams working with Scrum. As results, they reported that: (i) developers considered their own TD taxonomy based on the type of work they performed and their personal understanding of the term; (ii) developers pay more attention to design and test debt; and (iii) having dedicated teams to eliminate debt items during sprints is a good initiative to reduce TD.

Holvitie *et al.* [8] conducted a survey with professionals from Finland and found that most participants were familiar with the term TD. The authors also pointed out that more than half of the interviewees realized that practices directly related to software implementation have a positive effect on TD and its management. Finally, it was also identified that the project stage most affected by TD is the implementation, and the main cause for the occurrence of TD is an inadequate definition of its architecture.

More recently, Ernst *et al.* [7] reported the results of a survey with software engineers and architects. The authors found that architectural decisions are the most important source of technical debt. Furthermore, while respondents believe the metaphor is itself important for communication, existing tools are not currently helpful in managing the details.

Finally, Spínola *et al.* [12] investigated the level of agreement of software professionals with phrases of effect ("TD Folklore") on TD. The results of this study indicated that TD is an important factor in software project management and not simply another term for "bad code". The replication process of this study, as well as the results obtained will be discussed in the sequence.

III. TD FOLKLORE SURVEY REPLICATION

A. TD Folklore Survey

The goal of the research conducted by Spínola *et al.* [12] was to evaluate a set of folklore statements about TD. For that, a survey was conducted with professionals in the area of software engineering. Its goal was to answer the research questions **RQ1** and **RQ2**.

The survey contains statements about TD, collected on websites, blogs and published articles. This list has only

TABLE I. TD FOLKLORE LIST

ID	TD Folklore Statement	Agreement	Consensus	Groups
1	Accruing technical debt is unavoidable on any non-trivial software project.	3	1	No tendency
2	Technical debt usually comes from short-term optimizations of time without regard to the long-term effects of the change.	4	2	Agreement and high to medium consensus
3	It is very difficult for software developers to see the true effect of the technical debt they are incurring.	3	2	No tendency
4	"Working off debt" can be motivational and good for team morale.	4	2	Agreement and high to medium consensus
5	The root cause of most technical is pressure from the costumer.	3	1	No tendency
6	Unintentional debt is much more problematic than intentional debt.	4	2	Agreement and high to medium consensus
7	The individuals choosing to incur technical debt are usually different from those responsible for servicing the debt.	3	1	No tendency
8	If technical debt is not managed effectively, maintenance costs will increase at a rate that will eventually outrun the value it delivers to customers.	4	1	Agreement and high to medium consensus
9	No matter what, the cost of fixing technical debt increases the longer it remains in the system.	4	2	Agreement and high to medium consensus
10	Paying off technical debt doesn't result in anything the customers or users will see.	2	2	Disagreement and medium consensus
11	The biggest problem with technical debt is not its impact on value or earnings, but its impact on predictability.	3	2	No tendency
12	Technical debt should not be avoided, but managed.	3	2	No tendency
13	Not all technical debt is bad.	3	2	No tendency
14	All technical debt is intentional.	1	1	Strongly disagreed

statements based on personal opinions and experiences without any evaluation.

The survey was structured into two sets of questions. The first one aims to establish the level of knowledge of the interviewees about software development and TD. In the second one, the survey contains 14 sentences (see Table I) and, for each of them, the authors asked the participants to indicate their level of agreement. The questionnaire used the 5-point Likert scale to indicate the level of agreement: "1: strongly disagree" to "5: strongly agree". In addition, participants had the option "I do not know". The survey was designed to be answered in about ten minutes.

To perform the data analysis, for RQ1 (agreement), the authors computed the median as indicator for central tendency. Thus, a median of 4 or 5 shows tendency towards agreement on a statement. On the opposite side, values of 1 and 2 indicate a tendency towards disagreement. For RQ2 (consensus), the authors calculated the spread in the distribution of responses for each statement by computing the inter quartile range (IQR). An interval size value of 1 indicates a low spread and high consensus. On the contrary, higher values show more spread and indicate less common opinion among participants.

B. Survey Replication

This section details how we evolved the initial design by adding RQ3, and how we planned and performed the replication of the study.

1) Procedure

To replicate the survey, it was not necessary to make any change in the original questionnaire. The main differences between the study and its replication rely on its:

- analysis methodology for RQ3 (discussed in Section III.B.3);
- population: differently from the original study that focused on practitioners, our replication has focused on both participants with experience and those with none prior experience on software development activities.

2) Data collection and subject characterization

We replicated the questionnaire in undergraduate and graduate software engineering classes with participants of differing expertise and background. Not all participants had experience with software development, however, theoretical concepts were presented in the software engineering discipline. Prior to the application of the questionnaire, the basic concepts of TD were presented to ensure that everyone knows the term. The concepts were carefully presented by the last author in order to do not affect the perception of the participants regarding the list of folklore statements.

In total, 70 participants (see Table II) answered the replicated questionnaire and the average time to complete it was 15 minutes. The survey participants were also asked for their target degree and years of experience, as well as the roles they had taken in software projects. Almost half of them (36) do not have experience with software development. Among the other 34 participants that have some experience on software development activities, most of them were developers (29), followed by project managers (6) and requirements analyst (6).

TABLE II. SUBJECTS´ CHARACTERIZATION

Role	[12]	Replication	Total
Developer	29	29	58
Project Manager	9	6	15
Requirement Analyst	2	6	8
Tester	4	1	5
Architect	3	0	3
Operations	1	0	1
Maintainer	1	0	1
Database Administrator	0	1	1
Academic Degree	**[12]**	**Replication**	**Total**
Undergraduate Student	2	52	54
Bachelor in Comp. Science	2	0	2
Graduate Student	14	0	14
Master Student	1	16	17
PhD Student	1	2	3
Undefined	17	0	17
# Experienced / no Experienced Subjects	**[12]**	**Replication**	**Total**
# Subjects with Soft. Exp.	37	34	71
# Subjects with no Soft. Exp	0	36	36
Years of Software Experience	**[12]**	**Replication**	**Total**
Mean	4.8	4.8	4.8

The mean time of experience for those who have software experience was 4.8 years (coincidently, the mean time of experience was the same considering both data sets). We can also see on Table II that most of participants (54) are undergraduate students, followed by master students (14) and PhD students (2).

Finally, by analyzing the whole population scenario (107 participants), we can notice that the most of participants have some experience as developer (58), followed by project managers (15) and requirements analyst (8). We can also notice that we have approximately 2/3 of participants with (71) and 1/3 of them without (36) prior experience on software development.

3) *Analysis methodology*

The research questions RQ1 and RQ2 were analyzed considering the whole dataset, including the data collected in [12]. Besides, the same methodology (based on median for RQ1 and inter quartile range for RQ2) considered by [12] was applied. In order to address RQ3, initially, we divided the whole dataset into two subsets representing participants with (71 subjects) and without (36 subjects) experience on software development. Our approach was twofold.

First, we computed the median and IQR values for each statement of each subset. Then, we compared differences between median and IQR values for three subsets (all participants indistinctly, and more and less experienced participants). In other words, we observed if the agreement and consensus were similar for each subset. A significant difference on agreement and consensus between these subsets evidences that the experience impacts on the level of agreement for that statement.

Second, we statistically compared the Likert scale values filled in by the participants in different subsets (more and less experienced participants). We adopted the Shapiro-Wilk normality test. For all cases, the distribution was not normal. Due to this, we adopted the Mann-Whitney, a non-parametric alternative to t-test, with a 0.05 p-value, to statistically test our hypothesis. The null hypothesis (H0) is: for a specific TD Folklore, there is no difference of the Likert scale values between more and less experienced participants. Rejecting the null hypothesis (p-value<0.05) evidences that the experience impacts on the TD Folklore level of agreement.

We then considered the two evidences in our analysis: i) the differences on the agreement and consensus; and ii) the statistical Mann-Whitney p-value. When these outcomes presented some inconsistence, we graphically analyzed the distribution of the Likert scale values.

IV. RESULTS AND DISCUSSION

This study surveyed participants with and without software development experience to investigate: (i) what statements from a TD Folklore list the participants agree with; (ii) what is the consensus around the statements collected about TD, and; (iii) if the perception on TD is similar or different between participants with and without experience on software development.

A. *Agreement with statements and Consensus*

Results of both research questions RQ1 and RQ2 are presented and grouped in Table I by central agreement tendency and consensus. Initially, we can see that no single folklore statement was commonly strongly agreed with. This indicates that none of the folklore statements were considered to be universally true. On the other side, there was one folklore statement that was commonly strongly disagreed with (#14). This result (i) suggests that software engineers are aware that there might be unknown TD items in their projects, and (ii) supports the ongoing line of research into tools that analyze source code for unknown debt. In this replication, we can also notice that we did not have higher values for IRQ, all values were close to 1. Thus, in general, there was a low spread and high consensus among the participants for this statement.

Some statements (#2, #4, #6, #8, and #9) presented a median of 4, which indicates a tendency towards agreement and high to medium consensus. These results indicated that there is a common belief that TD is an important part of software management. On the other hand, the statement #10 received general disagreement and medium consensus. This result indicated that from the point of view of participants, the presence of TD items could bring some impact for system users. Finally, seven other statements (#1, #3, #5, #7, #11, #12, and #13) showed no tendency on either side of the scale.

Fig. 1 complements the results of the analysis by median presented on Table I. In this figure, the percentages represent the distribution of answers according to the Likert scale. We can see that there is an inclination of the graph to the right side, indicating agreement (agree or strongly agree) on the folklore statements. By observing this graph, we also can highlight some statements (#2, #4, #5, #6, and #8) that reached a

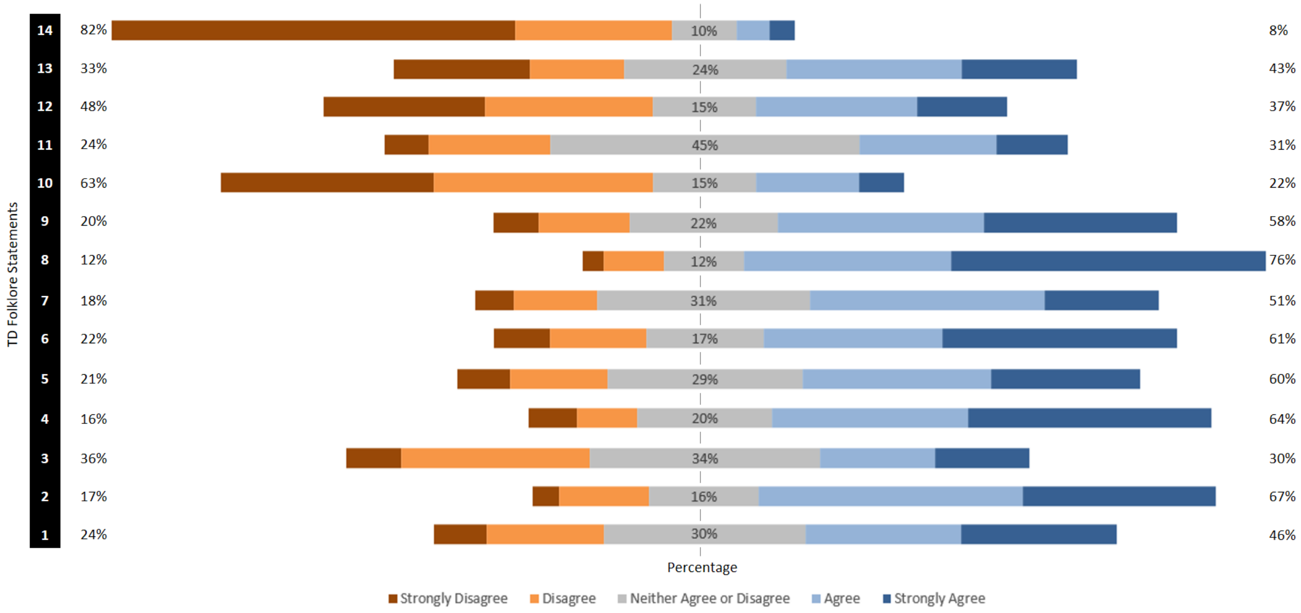

Figure 1. Agreement tendency analysis. Distribution of answers according to the likert scale

widespread agreement (>60% of answers are strongly agree or agree). On the other side, the participants clearly disagreed (>60% of answers are strongly disagree or disagree) with statements #10 and #14. Finally, despite the median analysis did not indicate an agreement tendency regarding the statements #5 and #7, most of the participants agreed with them.

B. *Do participants with and without software development experience have the same perception on TD concepts?*

Results of RQ3 are presented in Table III by central agreement tendency and consensus grouped by the participants' experience, and the Mann-Whitney p-value (last column).

The light gray lines represent the cases where agreement and consensus for both experienced and no experienced participants are similar to the values considering all participants indistinctly, as presented in Table I. Moreover, for these cases, it was not possible to reject the null hypothesis (Mann-Whitney p-value>0.05). The results evidence that the agreement with the statements was not impacted by the participants' experience for the statements #1, #2, #3, #4, #6, #8, #9, #10, #13 and #14.

TABLE III. AGREEMENT TENDENCY, CONSENSUS AND SHAPIRO-WILK P-VALUE GROUPED BY THE PARTICIPANTS' EXPERIENCE

ID	Experienced Participants		Participants without experience		Mann-Whitney
	Agreement	Consensus	Agreement	Consensus	p-value
1	3	3	3	1	0.3635
2	4	1	4	2	0.2789
3	3	2	3	2	0.5785
4	4	2	4	2	0.9725
5	3	2	4	2	0.0204
6	4	2	4	3	0.9719
7	**3**	**1**	**4**	**1**	**0.0722**
8	4	1	4	1	0.7241
9	4	2	4	2	0.2551
10	2	2	2	2	0.2503
11	**3**	**1**	**3**	**1**	**0.0107**
12	3	2	2	2	0.0219
13	3.5	2	3	3	0.0789
14	1	1	1	1	0.9574

The dark gray lines represent the cases where agreement, consensus, and the Mann-Whitney p-value (<0.05) evidence that the agreement with the statements was impacted by the participants' experience. It occurred only for the statements #5 and #12.

For the statements #7 and #11, we found inconsistencies between agreement/consensus and the hypothesis test. In order to better understand these cases, we show the distribution of the participants' agreement with the statements in Fig. 2. Statement #7 has similar distribution considering both group of participants. In opposition, is evident that values for statement #11 are higher for experienced than for no experienced participants. These results reinforce the Mann-Whitney p-value presented in Table III: the participants' experience impacts the agreement with the statement #11, and it does not impact the agreement with the statement #7.

Overall, we observed that the experience impacted only the

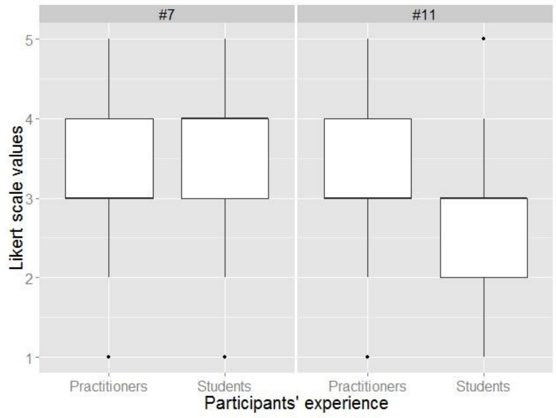

Figure 2. Agreement distribution for the Folklores #7 and #11

results for three of the statements (#5, #7, #12). For us, this indicates that the experience does not have significant impact on the TD Folklore analysis. However, this also evidences that we should not to consider all TD Folklores in the same way. For example, observing TD Folklores #5, #7, #12, we conjecture that the experience might be impacted the results because concerns about customer pressure (#5), decisions about who will pay the debt off (#7) and if the debt should be avoided (#12) seem to be a reflex of the scenarios faced by development teams in their daily activities. We are planning investigate these aspects in the future.

V. Threats To Validity

In this section, we discuss some threats to validity:

External validity. The participants of this replication were graduate and undergraduate students. One aspect mitigates the threat: in total, most participants had some professional software development experience. As can be seen in Table II, there are 71 participants with some software development experience against 36 participants without software development experience. Despite this, the results might not generalize to a context in which developers have long years of experience (15 to 20 years, for instance). However, it is important to note that our findings are based on a comparison between two groups of participants that are clearly distinct regarding to the level of professional experience.

Internal validity. Another threat to the validity was the possibility of the presentation made on TD before the distribution of the questionnaire influence the responses of the participants. To deal with this threat, the TD concepts were carefully presented by one of the authors of this work.

Construct validity. Likert scales assume that participants can accurately map their answers to a question into one dimension (e.g., strongly agree or disagree). Since TD is a complex concept, this may not be realistic in some cases. The TD Folklore statements investigated in this work may not be 100% mutually exclusive and exhaustive.

VI. Concluding Remarks

In this paper, we presented the results of a replicated survey on TD folklore. We revisited the original work from Spínola *et al.* [12] by expanding the population (from 37 to 107 respondents) and reviewing the results concerning the agreement/disagreement tendency and consensus about each folklore item. Besides, we also investigated if experience with software development affects the perception of the participants about the considered statements.

Regarding agreement/disagreement tendency and consensus, our results reinforce the findings previously reported in [12]. Thus, the results provide some evidence and motivation for exploring the following issues in TD research: (i) impact of TD management on maintenance costs (#2, #8, #9), (ii) relationship between servicing the debt and team motivation (#4), (iii) relation between unintentional and intentional debt impact on software projects (#6), (iv) how the impact of debt items increases (or decreases) during software evolution (#9), (v) prediction or estimation models for TD impact (#8, #9), (vi) impact of paying TD items off on customers (#10), and (vii) development of strategies to identify unintentional TD items (#14). Finally, the results suggest that software development experience does not interfere in the perception on the TD concept for the most cases.

In our future research agenda, we intend to combine the evidence identified in this work with new theories and empirical studies developed by our research group. Specifically, we intend to investigate causes and impacts of TD on software projects.

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 RESCUER project Grant: 490084/2013- 3.

References

[1] N.S.R. Alves, R.S. Araújo, and R.O. Spínola. A Collaborative Computational Infrastructure for Supporting Technical Debt Knowledge Sharing and Evolution. In: Americas Conference on Information Systems, 2015, Puerto Rico.

[2] N.S.R Alves, T.S. Mendes, M.G. Mendonça, R.O. Spínola, F. Shull, and C. Seaman. Identification and management of technical debt: A systematic mapping study. Information and Software Technology, v. 70, p. 100-121. 2016. DOI: https://doi.org/10.1016/j.infsof.2015.10.008

[3] F. Shull, V. Basili, J. Carver, J.C. Maldonado, G.H. Travassos, M. Mendonça, and S. Fabbri. 2002. Replicating Software Engineering Experiments: Addressing the Tacit Knowledge Problem. In Proc. of the 2002 Int. Symp. on Empirical Software Engineering, USA.

[4] Z. Codabux and B. Williams. Managing technical debt: An industrial case study. In 2013 4th International Workshop on Managing Technical Debt (MTD), (pp. 8-15). IEEE.

[5] W. Cunningham. The WyCash portfolio management system. In ACM SIGPLAN OOPS Messenger (Vol. 4, No. 2, pp. 29-30). ACM. 1992.

[6] A. Erickson. 2009. Don't "Enron" Your Software Project. Retrieved from http://www.informit.com/articles/article.aspx?p=1401640.

[7] N.A. Ernst, S. Bellomo, I. Ozkaya, R.L. Nord, and I. Gorton. 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.

[8] J. Holvitie, V. Leppanen, and S. Hyrynsalmi. (2014), Technical Debt and the Effect of Agile Software Development Practices on It - An Industry Practitioner Survey. In MTD 2014, pp. 35-42.

[9] T. Klinger, P. Tarr, R. Wagstrom, and C. Williams. "An enterprise perspective on technical debt," in Proceedings of the 2nd Workshop on Managing Technical Debt. ACM, 2011, pp. 35–38.

[10] E. Lim, N. Taksande, and C. Seaman. "A balancing act: what software practitioners have to say about technical debt," Software, IEEE, vol. 29, no. 6, pp. 22–27, 2012.

[11] W. Snipes, B. Robinson, Y. Guo, and C. Seaman. Defining the decision factors for managing defects: a technical debt perspective. In 2012 Third Int. Work. on Managing Technical Debt, (pp. 54-60). IEEE.

[12] R.O. Spínola, N. Zazworka, A. Vetrò, C. Seaman, and F. Shull (2013). Investigating technical debt folklore: Shedding some light on technical debt opinion. In Proc. of the 4th Int. Work. on Managing Technical Debt

[13] N. Juristo and S. Vegas. Using differences among replications of software engineering experiments to gain knowledge. In Proceedings of the 2009 3rd Int. Symp. on Empirical Soft. Eng. and Measurement. IEEE Computer Society, Washington, DC, USA, 356-366.

[14] 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.