POLITECNICO DI TORINO
Repository ISTITUZIONALE

Investigating Technical Debt Folklore - Shedding some light on technical debt opinion

Original
Investigating Technical Debt Folklore - Shedding some light on technical debt opinion / Spínola, R.; Zazworka, N.; Vetro', Antonio; Seaman, C.; Shull, F.. - STAMPA. - (2013), pp. 1-7. (Intervento presentato al convegno Fourth International Workshop on Managing Technical Debt tenutosi a San Francisco, USA nel May 20, 2013) [10.1109/MTD.2013.6608671].

Availability:
This version is available at: 11583/2506362 since:

Publisher:
IEEE

Published
DOI:10.1109/MTD.2013.6608671

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

12 October 2024

Investigating Technical Debt Folklore

Shedding some light on technical debt opinion

Rodrigo O. Spínola
Graduate Program in Systems and Computer
Universidade Salvador - UNIFACS
Salvador, BA, Brasil
rodrigo.spinola@pro.unifacs.br

Nico Zazworka
Elsevier Information Systems GmbH
Frankfurt, Germany
nico@zazworka.com

Antonio Vetro'
Dept. of Control and Computer Engineering
Politecnico di Torino
Torino, Italy
antonio.vetro@polito.it

Carolyn Seaman
Department of Information Systems
UMBC
Baltimore, MD, USA
cseaman@umbc.edu

Forrest Shull
Fraunhofer USA Center for
Experimental Software Engineering
College Park, USA
fshull@fc-md.umd.edu

***Abstract*— We identified and organized a number of statements about technical debt (TD Folklore list) expressed by practitioners in online websites, blogs and published papers. We chose 14 statements and we evaluated them through two surveys (37 practitioners answered the questionnaires), ranking them by agreement and consensus. The statements most agreed with show that TD is an important factor in software project management and not simply another term for "bad code". This study will help the research community in identifying folklore that can be translated into research questions to be investigated, thus targeting attempts to provide a scientific basis for TD management.**

***Index Terms*— technical debt; software maintenance.**

I. Introduction

The technical debt (TD) metaphor describes a tradeoff between short-term and long-term goals in software development [1]. Since many in the software community find the metaphor sound and intuitive, it has facilitated discussion between technical and non-technical stakeholders involved in the software development process, as well as between practitioners and researchers. The metaphor aids discussion by providing a familiar framework and vocabulary from the financial domain.

The acceptance and use of the TD metaphor is in large part because it is easily understood. But this also raises a concern. Because the TD metaphor is easy to understand, it is also easy to talk about, expand on, and relate experience to. A quick search of TD literature reveals subjective opinions, personal views, and catch phrases on such channels as blogs and online essays. While the scholarly literature on TD is increasing (thanks in part to the MTD workshops), there is a plethora of attention-grabbing pronouncements in cyberspace that have not been evaluated before they were published, often reflecting the authors' guesses and experience on the subject of TD.

This scenario, rich in different and sometimes contradictory but colorful opinions, but without any assessment, can lead to the emergence of *folklore*. According to the Cambridge Dictionary, the term folklore means traditional stories, beliefs, and customs of a group of people. Folklore can sometimes hide valuable information originating from people's experience that, if evaluated, could contribute positively to the study of the area. Thus, we believe that TD folklore needs further investigation. Our rationale is that if any folklore is either widely agreed to or widely disagreed with by a large group of knowledgeable people, then those propositions are more likely to be good candidates for future research. On the contrary, mixed responses to a statement of folklore can indicate that it is not commonly believed, depends on many factors, or that the statement itself is not yet formulated as precisely as needed. Commonly believed folklore can help researchers to gather ideas for theories, hypotheses, research questions, and follow-up experiments.

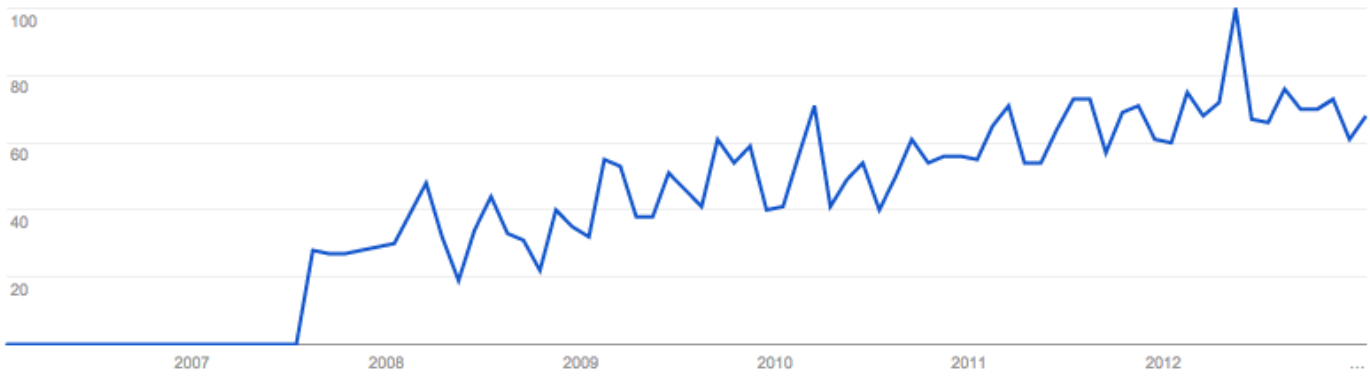

Figure 1: relative search volume for "Technical Debt" on Google.com

The first contribution of this paper is the identification and organization of common beliefs on TD (a TD Folklore list) expressed by practitioners and researchers in online websites, blogs and published papers.

As a second contribution, we evaluated the organized TD Folklore list through two surveys (37 practitioners answered the questionnaires). As a result, we reorganized the folklore list by rank of agreement and consensus.

This study will help us in understanding what practitioners have said about TD and what folklore, at this moment, seems to make sense and constitute good candidates for more detailed investigation.

Besides this introduction, this paper is organized as follows. Section 2 presents some background knowledge on the study of TD. Next, Section 3 discusses the goal, procedure, and data collection strategy that we followed in this study. Sections 4 and 5 present the results from executing the folklore survey and their interpretation. Section 6 discusses some threats to validity involved in the study, and finally, Section 7 provides the final conclusions to this work.

II. Background

Many advances have been made in the TD management area in recent years. Since 2010, the research community has been meeting annually at the Workshop on Managing Technical Debt. In the first 3 workshop editions comprehensive studies and positions were presented:

- to propose managing technical debt as a part of the future research agenda for the software engineering field [1];
- to discuss the management of technical debt, in particular focusing on assessing current practice in industry and to further refine a research agenda for software engineering in this area [2];
- to discuss managing technical debt as a part of the research agenda for the software engineering field, in particular focusing on eliciting and visualizing debt, and creating payback strategies [3].

IEEE Software also dedicated a Special Issue on the topic in 2012 [4]. In this special issue, topics like estimating the principal, methods to manage the technical debt, practitioners' views on technical debt, etc., were discussed.

But not only peer-reviewed papers are found in the literature in this area. There are also some interesting online websites and blogs, for example, ontechnicaldebt.com, blogs.construx.com, and blog.techdebt.org.

This mixed set of information contributes to the growth of the discussion around the subject. This rise in popularity is supported by search trends as reported by trends.google.com (Figure 1): over the past 5 years more and more Google users have been searching for the term "Technical Debt". Moreover, this rise in interest creates rich communication channels where practitioners can present their opinions without, often, any kind of evaluation. For example, it is common to find statements like "*not all technical debt is bad*" or "*all technical debt is intentional*" but, are these statements that are commonly believed? And, should researchers invest more time in investigating the universal truthfulness of these statements? Or, on the contrary, are these statements not commonly agreed on, and do we need to reformulate them?

On the next section we will present a study that starts to shed some light on this discussion.

III. Survey

The goal of the survey was to test a set of folklore statements about TD. These statements were identified by searching a number of online websites and blogs, as well as some published papers, for potential folklore statements, and then refining the list into the items presented in Table I. These statements might have been expressed by individuals or by groups, but most were not scientifically based[1].

The rationale of our survey method is that, if these statements evoke a consistent set of agreement or disagreement responses, then they are more likely to be good candidates for future research. On the other hand, mixed responses to a statement can indicate that it is not commonly believed, depends on other factors, or that the statement itself is not yet formulated as precisely as needed.

[1] Except one, originated from the paper "An Enterprise Perspective on Technical Debt" [5] and "To Pay or Not to Pay Technical Debt" [16].

TABLE I. TD FOLKLORE LIST

ID	TD Folklore statement	Reference
1	Accruing technical debt is unavoidable on any non-trivial software project	[6] [13] [16]
2	Technical debt usually comes from short-term optimizations of time without regard to the long-term effects of the change.	[7]
3	It is very difficult for software developers to see the true effect of the technical debt they are incurring.	[12]
4	"Working off debt" can be motivational and good for team morale.	[8]
5	The root cause of most technical debt is pressure from the customer.	[14]
6	Unintentional debt is much more problematic than intentional debt.	[5]
7	The individuals choosing to incur technical debt are usually different from those responsible for servicing the debt.	[5]
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.	[9]
9	No matter what, the cost of fixing technical debt increases the longer it remains in the system.	[9]
10	Paying off technical debt doesn't result in anything the customers or users will see.	[6]
11	The biggest problem with technical debt is not its impact on value or earnings, but its impact on predictability.	[10]
12	Technical debt should not be avoided, but managed.	[12]
13	Not all technical debt is bad.	[11] [13]
14	All technical debt is intentional.	[15]

A. Procedure

The study was organized in two phases: Identifying TD Folklore Statements and Evaluating TD Folklore Statements.

In the first phase, the first author conducted searches on the Internet looking for TD folklore statements. The search was performed on online websites, blogs, and published papers, and considered the first 100 results (stopping criteria) from Google search engine using the search key phrase "*technical debt"*. From those results, the researcher looked for TD Folklore statements (i.e. any statement that might be subject to opinion, or that might be a good candidate for further investigation) on all links that pointed to any, at least, small article or presentation on TD. In total, 16 online articles of different types were selected for data extraction (references [5] through [20]).

This initial list of folklore statements was then analyzed by four of the authors, including the one who compiled the initial list, in a discussion meeting. During this meeting, a list of 14 potential TD folklore statements (Table I) was selected to be used in the second phase of the study. The choice of statements for the final list was subjective. Each researcher indicated the statements they found most interesting. In some cases, statements from the original sources were reworded or combined for clarification.

For the second phase, Evaluating TD Folklore Statements, two surveys were conducted. The first survey (in the following referred to as the "**Online Survey**") was an online survey that was advertised during a webinar and on www.ontechnicaldebt.com. The second survey (in the following referred to as the "**Paper Survey**") was filled out on paper. In both cases, participants were invited to indicate their level of agreement for each of TD folklore statements as well as provide some software engineering background information.

The questionnaire used a 5-point Likert scale to indicate the agreement level: "1: strongly disagree", to "5: strongly agree". In addition, the Paper Survey participants were provided with the option of "I don't know", which was not available in the Online Survey. We will report on possible effects of this difference later.

The survey responses allow us to answer two research questions:

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

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

B. Data Collection and subject characterization

Thirty-seven participants filled in digital and printed versions of the questionnaire. The survey was executed in different contexts with participants of differing expertise and background:

- Online-Survey: In 2011 and 2012, two of the authors presented three webinars to Lockheed Martin/IEEE and Boeing on current research in automated identification of TD through code analysis tools. Webinar participants as well as readers of the corresponding blog article on www.OnTechnicalDebt.com[2] were invited to fill in the Online Survey and 17 responses were obtained in total.

TABLE II. SUBJECTS' PROFILE

Role (Online and Paper Survey, multiple answers possible)	
Developer	29
Project Manager	9
Tester	4
Architect	2
Requirement Analyst	2
Solution Architect	1
Operations	1
Maintainer	1
Academic Degree (Paper Survey only)	
Undergraduate Student	2
Bachelor in Computer Science	2
Graduate Student	14
Master Student	1
PhD Student	1
Years of Software Experience (Paper Survey only)	
Mean	4.8 years

[2] http://www.ontechnicaldebt.com/blog/identifying-and-measuring-technical-debt-ieee-software-boeing/

The level of detail on requested personal information was kept intentionally low (e.g., participants could report on their role but were not asked to report on their degree or years of experience) to increase the chance of participation and to keep the online survey as short as possible. Participants in the online survey are likely to have an increased interest in the TD metaphor as they participated in the webinar or browsed the OnTechnicalDebt website. The webinars were intended for a US audience, but the survey was on a website accessible world-wide. The online survey was in English.

- Paper-Survey: As part of another ongoing technical debt study, 15 students from a graduate course on Software Engineering and 5 participants from one of our industrial partners filled in the Paper Survey. All 20 of these participants completed the survey after a training session on technical debt led by one the authors. The training sessions for the 15 students and for the 5 practitioners was the same (same instructor, same slides), except that the students received the training in person during their class, and the practitioners' training was online (using a Skype video call). Both the training and the survey were in Portuguese for all Paper Survey participants. The Paper Survey questionnaire was a straightforward translation of the Online Survey. Paper Survey participants were also asked for their target degree and years of experience, as well as the roles they had taken in software projects.

Most of the participants of the Online Survey were developers (29), followed by project managers (9) and testers (4). Other roles[3] were negligible in quantity. From the group of participants of the Paper Survey all but two participants had completed their undergraduate computer science degree. Participants in the Paper Survey have approximately on average 5 years of experience in software development.

C. Analysis methodology

We performed analysis on the merged data set from both the Online and Paper Surveys, as well as on the individual data sets in order to study differences and reduce threats to validity from merging. An answer to a folklore statement was excluded if the participant chose to answer "I don't know" in the Paper Survey, or if the participant chose to not answer it in the online survey.

In order to address RQ1, we considered the 5-Point-Likert scale as an ordinal metric and hence computed the *median* as indicator for central tendency. A median of 4 or 5 shows tendency towards agreement on a statement. Values of 1 and 2 indicate a tendency towards disagreement. A value of 3 indicates no tendency to either side. If the median was a non-integer (i.e., not a whole number) we chose to use a conservative strategy, and rounded towards the central point of the Likert scale (e.g. 4.5 was rounded to 4, and 2.5 was rounded to 3).

As a measure for consensus (RQ2) we calculated the spread in the distribution of responses for each statement by computing the size of the interval between the 25th and 75th percentiles, also known as *inter quartile range* (IQR). For example, an interval size value of 1 indicates that 50% of the answers fall on two adjacent Likert scale values (e.g., 3 and 4; or 4 and 5), implying a low spread and high consensus. Higher values show more spread and indicate less common opinion among participants. In the following analysis and data presentation, we report the inter quartile range and label the spread as low (IQR=1), medium (IQR=2) and high (IQR=3), as these three values were the only values obtained in the results[4].

IV. Results and Interpretation

Results of both research questions are presented in Table III by central tendency and consensus. For each statement we show the results for data sets of the (P)aper survey, the (O)nline survey and the (M)erged result. For example, statement #14 obtained the same result, independent of which survey and data set is considered, P, O, or M. Statement #8 obtained different results in the P, O, and M data sets. The following observations can be drawn from the results.

No single folklore statement was commonly strongly agreed with in both surveys, but statement #8 was strongly agreed to in the online survey. This indicates that none of the folklore statements were considered to be universally true in any software project and that the rules and models for TD management might differ from one project to another or are simply not yet expressed in these folklore statements.

High disagreement with the statement that "All Technical Debt is intentional" (#14 in all datasets) indicates that many practitioners have been surprised to find TD that was not incurred intentionally. This strongly supports the ongoing line of research into tools that analyze source code for "hidden" and unknown debt.

Statements #2, #4, #8, #12 and #13 received general agreement and high to medium consensus (except, in some cases, in the Paper Survey), which suggests that these are good candidates for further scientific studies. These statements show a trend towards a common belief that TD is an important part of software management (statements #8, #12) and not simply another term for "bad code" (statement #13). Statement #13, indicating agreement that "not all TD is bad", also motivates investigation into that "sweet spot" between an acceptable and healthy level of debt, and a level that is approaching dangerous. Statement #4 is a very interesting candidate for future research on the interaction between social aspects in software development and TD management. If working off TD indeed improves team morale and motivation, then the effects of

[3] Participants could indicate multiple roles

[4] Theoretically the IQR values 0,1,2,3 and 4 can be obtained from a 5-Point-Likert scale, but they did not occur in our data set. In cases of non-integer IQRs we chose to be conservative and rounded up to indicate less consensus

TABLE III . TD FOLKLORE LIST BY RANK OF AGREEMENT AND CONSENSUS. THE INDICATORS IN PARENTHSIS SHOW DIFFERENT RESULTS FROM THE ONLINE SURVERY (O), PAPER SURVEY (P), AND THE RESULT FROM THE MERGING BOTH RESULTS (M).

Tendency/Spread	1 Low Spread Higher Consensus	2 Medium Spread	3 High Spread Lower Consensus
5 Strongly Agree	(O8) 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 Agree	(M2,O2) Technical debt usually comes from short-term optimizations of time without regard to the long-term effects of the change. (M4,P4) “Working off debt" can be motivational and good for team morale. (O6) Unintentional debt is much more problematic than intentional debt. (O7) The individuals choosing to incur technical debt are usually different from those responsible for servicing the debt. (M8) If technical debt is not managed effectively, maintenance costs will increase at a rate that will eventually outrun the value it delivers to customers. (M12,O12) Technical debt should not be avoided, but managed. (M13, O13,P13) Not all technical debt is bad.	(M1,O1) Accruing technical debt is unavoidable on any non-trivial software project. (P8) If technical debt is not managed effectively, maintenance costs will increase at a rate that will eventually outrun the value it delivers to customers.	(O4) “Working off debt" can be motivational and good for team morale. (P9) No matter what, the cost of fixing technical debt increases the longer it remains in the system.
3 Neither Agree or Disagree	(P1) Accruing technical debt is unavoidable on any non-trivial software project. (M7,P7) The individuals choosing to incur technical debt are usually different from those responsible for servicing the debt. (O9) No matter what, the cost of fixing technical debt increases the longer it remains in the system. (M11,O11,P11) The biggest problem with technical debt is not its impact on value or earnings, but its impact on predictability.	(P2) Technical debt usually comes from short-term optimizations of time without regard to the long-term effects of the change. (M3,O3,P3) It is very difficult for developers to see the true effect of the technical debt they are incurring. (M5,O5,P5) The root cause of most technical debt is pressure from the customer. (M9) No matter what, the cost of fixing technical debt increases the longer it remains in the system. (P10) Paying off technical debt doesn't result in anything the customers or users will see. (P12) Technical debt should not be avoided, but managed.	(M6,P6) Unintentional debt is much more problematic than intentional debt.
2 Disagree		(M10,O10) Paying off technical debt doesn't result in anything the customers or users will see.	
1 Strongly Disagree	(M14,O14,P14) All technical debt is intentional.		

paying back TD might have a larger positive impact on a software project than just increased maintainability.

Lastly, one folklore statement standing out in the merged data set with respect to level of consensus is the statement “Unintentional debt is much more problematic than intentional debt” (#6). Answers were widely spread with no clear tendency, indicating that different subjects have had very diverse experiences with intentional and unintentional debt in their projects. This shows that future research should not only focus on uncovering hidden (i.e. unintentional) debt but also cover managing the already known imperfections in software design, code, and documentation. It further suggests that

simply revealing TD in a project does not solve the problem, as known TD is still problematic.

V. Threats to Validity

We received a rather low number of responses (17 participants in the online survey, 20 participants in the paper survey), which results in limitations and the need for care when interpreting these numbers. Specifically, the data sets originate from two different surveys with a close-to-similar answer scheme but different collection methods: participants of the paper survey were not anonymous as were the participants of the online survey, and more background information was collected on the participants of the paper survey. To analyze the differences in results we performed two types of statistical analysis:

1. We tested if the answers of the two data sets were statistically equal or different on a 5% level (Mann-Whitney Tests)
2. We tested how similar the answers are with respect to central tendency (median) (Effect size computed as Cliff's delta[]).

Results of Mann-Whitney suggest that the amount of data is insufficient to show equality and difference with statistical certainty for most of the folklore statements. Results for effect size (Cliff's delta) indicate "negligible", "small", and "medium" differences when interpreted with [21]. Thus, we cannot strongly argue that the data sets must, or must not be, analyzed separately. To reduce the threat of data misinterpretations we present all three of the datasets separately (online, paper, and merged data).

Moreover, this small subset of responses originates from a potentially biased group of participants, i.e. subjects who have had unpleasant experiences with TD (and so were motivated to attend our webinar) or subjects who may have actually been the sources of some of the TD folklore statements (construct threat).

A further construct threat is introduced by the "don't know option" of the Paper Survey, which was not present in the Online Survey, where it was possible to not answer a question by not selecting one of the answers. We handled this threat by discarding those answers before merging the two datasets from the Paper and Online Surveys after careful analysis. Results showed that only a very small number of participants chose to not select an answer in the Online Survey (1 out of 238 answers) and to select the "I don't know" option in the Paper Survey (6 out of 280 answers).

Finally, the Paper Survey was translated into Portuguese, so language constraints and cultural idioms may have had an influence in the understanding of the statements. To deal with this, one of study's researchers is Brazilian and worked carefully to avoid or minimize any bias or misunderstanding in the translation process from English to Portuguese, and during the survey execution.

VI. Conclusion

This paper describes an investigation into TD "folklore". The authors compiled a list of 14 folklore statements, i.e. expressions of opinion and experience, taken from both scholarly and "grey" (i.e. blogs, online essays, websites, etc.) literature. We then tested the consensus around those statements by surveying software practitioners on their agreement with each folklore statement. The results provide some evidence and motivation for exploring the following issues in TD research:

- Methods and tools for finding unintentional, and therefore likely hidden, TD in source code and other artifacts;
- Methods and techniques for managing and tracking TD;
- Investigation of the "sweet spot" between an acceptable and healthy level of debt, and a level that is approaching dangerous;
- The relationship between TD and team morale and motivation;
- Exploring the differences, in both cause and effect, between intentional and unintentional TD.

Clearly there are many open questions to investigate and much work to be done by the research and practitioner community before we can provide industry with reliable advice about how to manage TD on software projects. The aim of the work described here is to help guide future research towards areas that would be of most interest and help to practitioners.

Acknowledgment

The participation of Seaman and Zazworka in this work is supported by the US National Science Foundation, award #0916699. The author Spínola would like to thank the Brazilian National Council for Scientific and Technological Development (CNPq) (award #201440/2011-3) and Fundação de Amparo à Pesquisa do Estado da Bahia (FAPESB) for their support of this research.

We also would like to thank Alexandra Szynkarski for promoting the online survey on OnTechnicalDebt.com and Marco Torchiano of Politecnico di Torino for his suggestions on the statistical analysis.

References

[1] N. Brown, Y. Cai, Y. Guo, R. Kazman, M. Kim, P. Kruchten, E. Lim, A. MacCormack, R. Nord, I. Ozkaya, R. Sangwan, C. Seaman, K. Sullivan, and N. Zazworka. 2010. Managing technical debt in software-reliant systems. In Proceedings of the FSE/SDP workshop on Future of software engineering research (FoSER '10). ACM, New York, NY, USA, 47-52.

[2] I. Ozkaya, P. Kruchten, R. L. Nord, and N. Brown. 2011. Managing technical debt in software development: report on the 2nd international workshop on managing technical debt, held at ICSE 2011. SIGSOFT Softw. Eng. Notes 36, 5 (September 2011), 33-35. DOI=10.1145/2020976.2020979.

[3] P. Kruchten, R. L. Nord, I. Ozkaya, and J. Visser. 2012. Technical debt in software development: from metaphor to theory report on the third international workshop on managing technical debt. SIGSOFT Softw. Eng. Notes 37, 5 (September 2012), 36-38. DOI=10.1145/2347696.2347698.

[4] P. Kruchten, R. L. Nord, I. Ozkaya, "Technical Debt: From Metaphor to Theory and Practice," IEEE Software, vol. 29, no. 6, pp. 18-21, Nov.-Dec., 2012.

[5] T. Klinger, P. Tarr, P. Wagstrom, and C. Williams. 2011. An enterprise perspective on technical debt. In Proceedings of the 2nd Workshop on Managing Technical Debt (MTD '11). ACM, New York, NY, USA, 35-38. DOI=10.1145/1985362.1985371.

[6] J. Atwood. 2009. Paying Down Your Technical Debt. Retrieved from http://www.codinghorror.com/blog/2009/02/paying-down-your-technical-debt.html. Retrieved on February 04, 2013.

[7] A. Erickson. 2009. Don't "Enron" Your Software Project. Retrieved from http://www.informit.com/articles/article.aspx?p=1401640. Retrieved on February 04, 2013.

[8] S. McConnell. 2007. Technical Debt. Retrieved from http://blogs.construx.com/blogs/stevemcc/archive/2007/11/01/technical-debt-2.aspx. Retrieved on February 04, 2013.

[9] A. Borton. 2012. Managing and Avoiding Technical Debt. Retrieved from http://www.docstoc.com/docs/140224021/Managing-and-Avoiding-Technical-Debt. Retrieved on February 04, 2013.

[10] J. Highsimth. 2010. The Financial Implications of Technical Debt. Retrieved from http://jimhighsmith.com/the-financial-implications-of-technical-debt/. Retrieved on February 04, 2013.

[11] G. Short. 2010. Credit Crunch Code. Retrieved from http://www.slideshare.net/garyshort/technical-debt-2985889. Retrieved on February 04, 2013.

[12] E. Allman. 2012. Managing Technical Debt. Retrieved from http://queue.acm.org/detail.cfm?id=2168798. Retrieved on February 04, 2013.

[13] D. Helper. 2011. Is it ok to have technical debt? Retrieved from http://agile.dzone.com/articles/it-ok-have-technical-debt. Retrieved on February 04, 2013.

[14] K. S. Rubin. 2012. Essential Scrum: A Practical Guide to the Most Popular Agile Process (1st ed.). Addison-Wesley Professional.

[15] Feyn Blog. 2012. Technical Debt. Retrieved from http://feyn.com/technical-debt/. Retrieved on February 04, 2013.

[16] F. Buschmann. 2011. "To Pay or Not to Pay Technical Debt" IEEE Software, vol. 28, no. 6, pp. 29-31, Nov.-Dec. 2011, doi:10.1109/MS.2011.150.

[17] Cunningham & Cunningham. 2012. Technical Debt. Retrieved from http://www.c2.com/cgi/wiki?TechnicalDebt. Retrieved on February 04, 2012.

[18] S. Palani. 2012. Managing The Technical Debt Risk. Retrieved from http://aroundthechaos.com/technical-debt-risk. Retrieved on February 04, 2012.

[19] M. Fowler. 2009. Technical Debt. Retrieved from http://martinfowler.com/bliki/TechnicalDebt.html. Retrieved on February 04, 2012.

[20] J. Mischel. 2012. In Praise of Technical Debt. Retrieved from http://blog.mischel.com/2012/01/07/in-praise-of-technical-debt/. Retrieved on February 04, 2012.

[21] J. Romano, Kromrey, J.D., Corragio, J., and Skowronek, J. 2006. Appropriate statistics for ordinal level data : Should we really be using t-test and cohen's d for evaluating group differences on the nsse and other surveys? In *Annual meeting of the Florida Association of Institutional Research.*