On the Relationship Between Technical Debt Management and Process Models

Nicolli Rios, Federal University of Rio de Janeiro

Sávio Freire, Federal University of Bahia and Federal Institute of Ceará

Boris Pérez, University of Los Andes and Francisco de Paula Stder University

Camilo Castellanos, University of Los Andes

Darío Correal, University of Los Andes

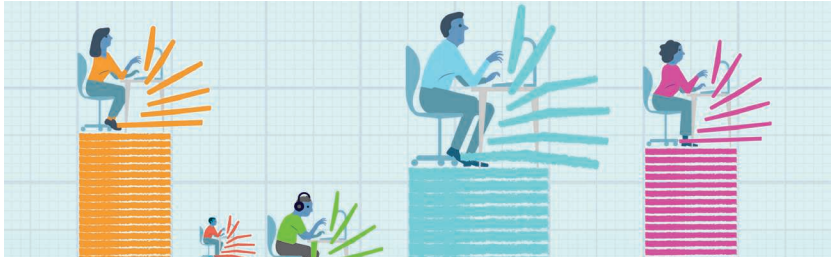
Manoel Mendonça, Federal University of Bahia

Davide Falessi, University of Rome "Tor Vergata"

Clemente Izurieta, Montana State University and Idaho National Laboratories

Carolyn B. Seaman, University of Maryland Baltimore Country

Rodrigo Spínola, Salvador University and State University of Bahia



// This study investigates the potential relationship between development process models and technical debt effects and their management by surveying 432 practitioners from software organizations in Brazil, Chile, Colombia, and the United States. //

TECHNICAL DEBT (TD) contextualizes the problem of taking design shortcuts in pending development tasks as a type of debt that brings a short-term benefit to the project, usually in terms of increased development speed or shortened time to market, but that may have to be paid with interest later on in the software development process.[1] As TD is potentially a result of poor decisions related to development tasks, one might expect that the inherent context resulting from following different process models, such as agile, hybrid, or traditional,[2] leads to differences in how teams view the effects of TD, and thus how they manage TD.

The effects of TD can impact software projects in several ways.[3] Having information about potential TD effects aids in the prioritization of TD items to pay off, by supporting a more precise impact analysis and the identification of corrective actions to minimize possible negative consequences for the project. It is even more useful to know what the common TD effects are in specific contexts, e.g., when following a particular process model, so that projects can choose TD

Digital Object Identifier 10.1109/MS.2021.3058652
Date of current version: 20 August 2021

 0740-7459/21©2021IEEE

repayment strategies that ameliorate effects that they are likely to experience, as opposed to effects that are less likely to apply to their project.

TD management encompasses activities related to TD prevention (applying good practices that minimize the occurrence of debt[4]), TD monitoring (of the costs and benefits of unresolved TD over time[5]), and TD repayment (the elimination of identified TD items[4]). One might expect that there are differences in how debt items are prevented, monitored or repaid, for different process models. For example, due to agile's focus on reacting to changes, practitioners following agile processes might be more able to prevent the occurrence of debt items. On the other hand, due to the high emphasis on documentation in traditional process models, debt repayment could be costier (as multiple artifacts are likely to be impacted by any change).

Although the aforementioned expectations and many others could be valid, the fact is that despite current efforts to understand TD effects,[3,6,7] and TD management,[1,4,5] very little is known about the relationship between TD effects, TD management, and process models. We approach this topic by surveying 432 practitioners from different countries in the context of the InsighTD Project[3] (see "The InsighTD Project"). Such investigation can provide guidance for practitioners in several ways:

- Understanding the impact of the process model on TD management could avoid, for example, "silver bullet" thinking: just by choosing a specific process model, the TD management would be trivial.
- Knowing the effects that are more prone to affect their processes would help the team narrow down the choices for TD management strategies, including repayment options, impact reduction, and risk management.
- Reducing the chances of making decisions based only on subjective opinions or assumptions about TD (that is, by too much only on their personal beliefs), practitioners might face several risks in their projects.[8]

THE INSIGHTD PROJECT

The InsighTD project started in 2017 and is based on an industry survey covering six main areas of investigation: TD concept, causes, effects, prevention, monitoring, and repayment. Previous InsighTD publications have addressed the following topics:

- What are the effects of different types of TD?
- When, how, and why do practitioners pay off TD in their projects?
- What do teams do to prevent TD and how successful are they?

Until recently, the InsighTD data set has not been large enough to partition according to the context questions asked in the survey (such as process model). This paper represents the first opportunity to explore differences between different types of projects in the InsighTD data. See http://www.td-survey.com/publication-map/ to learn more about other InsighTD investigations.

The Survey

The work discussed herein uses a subset of 14 questions from the InsighTD questionnaire. Q1–Q7 characterize the survey participants and their work context. In Q8, we presented the definitions of agile, hybrid, and traditional process models[2] and asked the participants which one was followed by the development team. Next, we ask participants to define TD in their own words in Q10. Then, we present a TD definition adapted from McConnell.[9] Q13 asks for an example of a TD item. Q22, Q24, and Q26 are yes/no questions about TD prevention, monitoring, and repayment, respectively. Finally, Q20 is an open-ended question about effects of TD. The answers to this question were analyzed qualitatively using the open coding technique[3] by at least three researchers.

Only responses from participants who provided a valid definition of TD for Q10 and a valid example of a TD item in Q13 were considered for analysis.[3] Thus, the survey obtained 432 valid answers from developers in Brazil (107), Chile (92), Colombia (134), and the United States (99). Most respondents identified themselves as proficient (33%), followed by competent (29%), expert (26%), beginner (11%), and novice (1%), indicating that, in general, the questionnaire was answered by professionals with experience in their functions.

According to Q8, 44% of the respondents followed hybrid, 41% followed agile, and 14% followed traditional process models. We first

checked for associations between the process model type and possible confounding factors (company size, team size, participant's role, experience level of participants, system size, and system age) using Pearson's Chi-squared test and Fisher's exact test. This check did not find any significantly significant effects. Thus, we can conclude that our data set does not have significantly different distributions of context factors for the three process models. If the reader is interested on additional details about the demographics, analysis procedures, and their execution, please access the supplementary material that accompanies this article at https://doi.org/10.1109MS.2021.3058652.

Is TD Management Different in Agile, Hybrid, and Traditional Process Models?

TD management encompasses a number of activities,[4,5] including prevention, monitoring, and repayment. The InsighTD survey addresses the TD management topic by asking participants about these three activities specifically through yes/no questions.

We used descriptive statistics and hypothesis testing to investigate the association between process model and how practitioners dealt with TD prevention, monitoring, and repayment. First, we divided the data set into three groups by process model (agile, hybrid, or traditional) along two dimensions (yes/no) for each TD management question. Then, we calculated the number of yes and no answers in each group for each question Q22 (prevention), Q24 (monitoring), and Q26 (repayment). Finally, we investigated whether there are significant differences between the groups using Pearson's Chi-squared statistical test and the V-Cramer statistic.

Figure 1(a) shows the percentage of each group indicating whether the TD item could be prevented. The cells of the table show the percentages of respondents, with actual totals per line and per column. Clearly, the distributions of answers are not different among the three process models, as confirmed by the V-Cramer statistic (0.0437) and the Pearson's Chi-squared statistical test (p-value 0.663). No matter which process model the development team was following, most practitioners thought that the TD could have been prevented.

This result is interesting because of the perception that agile's focus on reacting to changes would provide more opportunities to prevent debt. But these data show that practitioners following a traditional model still see opportunities for TD prevention. Clearly there is room for process improvement toward the prevention of TD, no matter which process model is being followed.

Figure 1(b) and Table 1 show the results for the question about TD monitoring, and indicate larger differences, especially between agile and traditional projects. Participants that follow agile process models are more inclined to monitor TD in comparison to those using traditional process models.

This result has two immediate implications. First, the effects of TD are commonly related to internal quality issues and many of them (for example, low internal quality, low maintainability, rework) are directly related to coding activities.[3] As coding occupies more attention and time in agile projects (as compared to traditional), practitioners on agile projects would also more immediately benefit from TD monitoring. Second, given the increased planning and monitoring burden in traditional processes (as compared to agile processes), TD monitoring might be seen as an unnecessary addition to that burden.

Figure 1(c) shows the percentage of each group (process model) indicating

(a)

	TD Prevention		
	No	Yes	
Agile	11%	89%	179
Hybrid	8%	92%	191
Traditional	10%	90%	62
	42	390	432

(b)

	TD Monitoring		
	No	Yes	
Agile	38%	62%	179
Hybrid	49.7%	50.3%	191
Traditional	58%	42%	62
	42	390	432

(c)

	TD Repayment		
	No	Yes	
Agile	55%	45%	179
Hybrid	62%	38%	191
Traditional	63%	37%	62
	42	390	432

FIGURE 1. The relationship between the process model and TD (a) prevention, (b) monitoring, and (c) repayment.

if the TD item was paid off. While there are some differences among groups, they are not large or significant (V-Cramer statistic 0.0718, Chi-squared *p*-value of 0.329). TD repayment did not happen in most cases regardless of the process model they followed.

In conclusion, although the overall tendency is not to repay the existing TD, regardless of the process model, we found a substantial percentage of items that were repaid (a mean of 40% of the cited items). Specifically, participants that follow agile process models seem to be slightly more prone to eliminate TD. This result can also be explained by the fact that debt items are more commonly found or reported in coding related activities. Those activities are subject of much focus in agile processes and, thus, the benefits of repaying the debt would be felt more immediately in agile process. On the other side, 63% of participants from traditional processes indicated that the debt items were not repaid. This high percentage indicates that the traditional process tends to suffer more from the negative effects of the presence of TD, and add up to well-known problems faced by traditional processes, in an equation that is hard to solve.

Finally, we noticed advances in terms of TD monitoring, which was performed for most of the cases (54% overall). However, the TD monitoring is significantly influenced by the followed process model. Agilists are more likely to monitor TD than hybridists and traditionalists.

Are the Effects of TD Different in Agile, Hybrid, and Traditional Software Projects?

Through qualitative analysis of the responses to Q20,[3] we identified 79 effects of TD in software projects. We then divided the cited effects into agile, hybrid, and traditional subsets based on Q8. The effects in each subset were ordered according to their frequency, that is, the number of participants who said that an effect was an impact of the TD example they were describing.

To better understand the relationship between TD effects and process models, we quantitatively measured the similarity of the effect ranking lists for each process model using rank-biased overlap (RBO). This analysis indicates whether, in general, the process models suffer from the same types of TD effects, and with the same frequency. RBO supports top-weighted ranked lists, thus, the first elements in a list have more impact on the similarity index than the latter ones.[10]

Figure 2 shows the RBO comparison between the ranked lists of effects identified from each process model. Each line of the figure represents a comparison between two of the process models. As the *p*-value increases, more elements of the lists are included in the comparison. The RBO indicates that the lists are overlapping when its value is closer to 1. As one might expect, the largest differences are found between the traditional and agile process models. The hybrid model had more similarity to both the agile and traditional models than they did to each other, although hybrid was much more similar to the traditional than the agile. Thus, we conclude from Figure 2 that agile software development projects experience the effects of TD in different ways than nonagile projects.

Still analyzing Figure 2, we can also observe that as RBO includes more effects in the comparison (*p*-value increases), the overlapping between the process models decreases, revealing that the most commonly cited effects on each list are very similar. Table 2 presents the five most commonly cited effects by process model, which are very similar. Thus, just looking at the most common effects is not enough to really understand the differences between the process models. The real difference is not in what effects are high on the list, but the balance or relative probabilities of the different effects. For instance, although delivery delay is the most common effect in all process models, it is more commonly faced in traditional models (25.8% of the cases) when compared with agile models (20.1%).

Table 1. The *p-value* and effect size between process models with respect to TD monitoring.

	Pearson's Chi-squared statistical test (with 95% of confidence level)		V-Cramer	
Process Model	*p-value*	Is there significant difference?	Value	Effect
Agile, Hybrid and Traditional	0.009421	Yes	0.1128245	Moderate
Agile and Traditional	0.005949	Yes	0.1675978	Strong
Agile and Hybrid	0.02291	Yes	0.1128245	Moderate
Hybrid and Traditional	0.2543	No	0.06247666	Weak

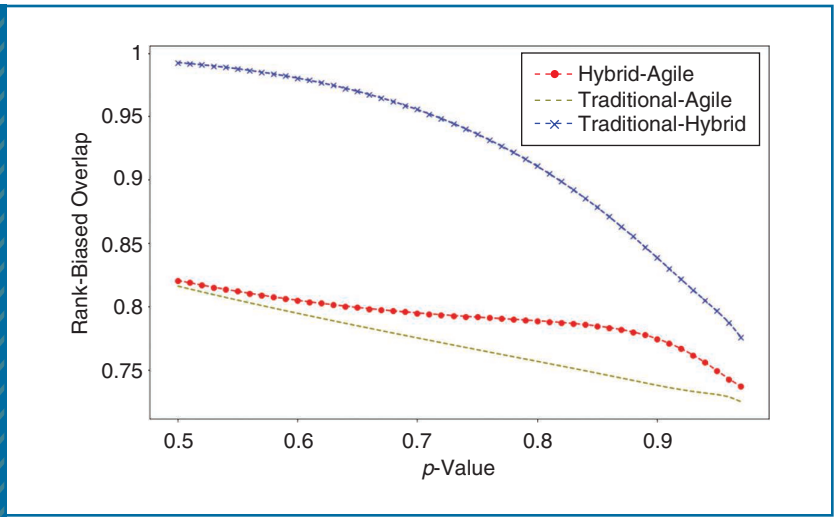


FIGURE 2. The RBO of effects ranks per process model.

Table 2. The top five TD effects cited frequency by the process model.

Ranking	Process Model		
	Agile	Hybrid	Traditional
1st	Delivery delay (20.1%)*	Delivery delay (24.6%)	Delivery delay (25.8%)
2nd	Low external quality (17.3%)	Low maintainability (18.3%)	Low maintainability (21%)
3rd	Low maintainability (16.8%)	Rework (16.2%)	Rework (16.1%)
4th	Rework (16.2%)	Low external quality (15.2%)	Low external quality (12.9%)
5th	Increased effort (8.9%)	Increased cost (6.8%)	Increased cost (9.7%)

**The number in parentheses represents the percentage of the number of citations of each effect by process.*

These findings can assist in decision making, incorporating a wider range of potential consequences in decision models that attempt to capture the long-term cost of TD. Although the overlapping of agile, hybrid, and traditional process models reveals that they share the majority (65%) of identified effects, these effects can impact projects slightly differently depending on which process model they are following. Thus, new approaches to mitigate TD effects should consider how frequently each effect impacts the process model under use. A way to incorporate this probability information into decision making is with the probabilistic diagrams of effects of TD, which are discussed in the next section.

For those interested on the definitions of the most commonly cited effects, and the full list of effects, please access the supplementary material that accompanies this article on IEEE *Xplore*.

Probabilistic Diagrams of Effects of TD

The probabilistic diagrams of effects of TD, proposed by Rios et al.,[11] highlight the most common effects that occur as a result of a problem, helping them to identify effects that they would not have identified otherwise. Such diagrams can support TD effect analysis meetings, in which the effects of TD items can be analyzed to support the definition of action plans to deal with them.

We can specialize the diagrams of the effects of TD to show only the effects related to a specific development process model. This is useful because there are differences on how frequently each effect impacts software projects following specific process models. For example, instead of having a general effect diagram that would indicate that increased cost is an effect of TD that can impact the project, by using specialized diagrams, an agile development team, for example, could see that they are less likely (5%) to experience this effect than a traditional (9.7%) process. Figure 3 presents such a specialized diagram for an agile process model (see the other diagrams in the supplementary material that accompanies this article on IEEE *Xplore*). The diagram is created based on the ranked list of effects for agile process models. It shows how the probabilities for each possible effect impact development teams and represents the effects using gray tones, where effects with higher probability are shown closer to the center and in darker tones.

Suppose that a team is using the diagram in Figure 3 to plan steps to mitigate the impacts of TD. One strategy would be to examine Figure 3,

beginning with the Planning and Management category and with the effect delivery delay, and then work their way up to reduction in scope. For each effect, the team would decide 1) whether that effect applies to their project and 2) whether it is possible to work on actions that reduce this effect. After going through all of the Planning and Management effects, the team could then move on to the Internal Quality Issues category, following the same process. Continuing in this way, the team could reflect on each category, until they have compiled a sufficient list of potential actions to minimize the impact of the presence of TD, or until they run out of time. In this way, a team can prioritize their time and improve the use of resources to focus on those effects most likely felt by the project, and whose elimination would be most likely to have a positive impact on it. Using a probabilistic diagram that is specialized to practitioners' context, in particular the process model they are following, makes this process more efficient and effective.

After using the diagrams, development teams will have a list of effects to be monitored. That list is the starting point to plan actions to minimize the impact or eliminate the debt. To this end, practitioners can adopt TD repayment practices. For example, code and design refactoring are commonly used practices to eliminate debt items for the category internal quality issues in agile process models. A comprehensive list of repayment practices related to each of the category of effects of the diagram is presented in Freire et al.[12]

Interested readers can find a detailed description on the use of the diagrams and their benefits in Rios et al.[11]

Threats to Validity

We sought to reduce conditions that limit our ability to generalize the

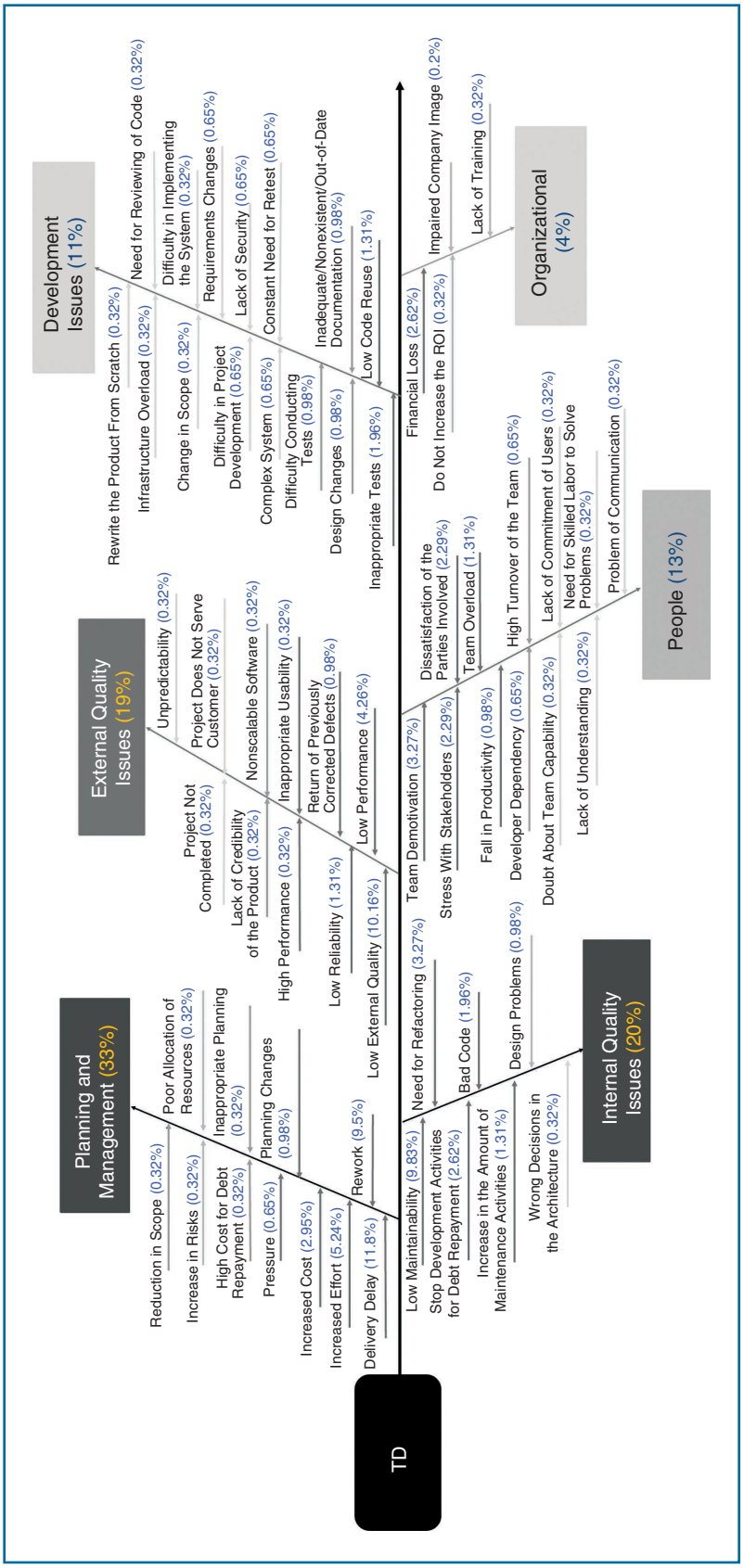


FIGURE 3. A probabilistic diagram of effects of TD for agile process model.

results by achieving a diversity of surveyed participants. The number of participants also reduces the chances of biased results due to specific groups of participants.

Also, participants may act differently than they usually do because they are part of a study. To prevent it, we clearly explain the purpose of the study and ask the participants to answer questions based on their own experience. We also explicitly stated that the questionnaire is anonymous and that the collected data are analyzed without taking into consideration the participants' identities.

The interested reader can find more details on the threats to validity in the supplementary material that accompanies this article on IEEE *Xplore*.

Takeaways for Practitioners

InsighTD data indicate that the current practice in TD monitoring and repayment is still far from ideal. However, practitioners working in agile projects are more likely to invest in TD monitoring and repayment activities, signaling that TD management might be improved by investing in some agile practices, such as iterative development and tightly knit teams. There is also a common perception, cutting across process models, that TD can be prevented. Thus, investing on prevention initiatives is likely to be supported by practitioners in all types of projects.

InsighTD results also show that the most common TD effects are felt to different degrees in agile versus. nonagile projects, so prioritization strategies for TD mitigation activities should also differ. We have presented a mechanism for using this information, probabilistic diagrams of TD effects, to make better decisions about anticipating and avoiding the effects of TD. To illustrate the benefit of using diagrams specialized by process models, consider the following scenario. Suppose that an agile project team is using a probabilistic diagram to plan TD repayment and other improvement activities. Suppose also that the characteristics of their customer base dictate the need to deliver very high external quality. If this project team was using a generic probabilistic diagram based on data from traditional projects, then according to Table 2, the diagram would indicate that the main benefits of repaying the debt would be to mitigate the effects delivery delay, low maintainability, and rework. Thus, the team would be led to believe that TD repayment would not contribute toward their primary goal, high external quality, and they would make decisions that deprioritize TD repayment. The impact on the project would be to ignore TD that leads to low external quality, raising the risk of affecting their top priority issue negatively. Thus, practitioners should use techniques based on historical data (such as probabilistic diagrams) that are specialized to the attributes of the current project, in particular the development process.

Acknowledgment

This article has supplementary downloadable material available at https://doi.org/10.1109MS.2021.3058652, provided by the authors.

References

1. C. Seaman and Y. Guo, "Measuring and monitoring technical debt," *Adv. Comput.*, vol. 82, pp. 25–46, 2011. **[Online]. Available: https://doi.org/10.1016/B978-0-12-385512-1.00002-5**
2. G. Theocharis, M. Kuhrmann, M. J. Münch, and P. Diebold, "Is water-scrum-fall reality? On the use of agile and traditional development practices," in *Proc. 16th Int. Conf. Product-Focused Softw. Process Improvement*, 2015, pp. 149–166. doi: 10.1007/978-3-319-26844-6_11.
3. N. Rios, R. O. Spínola, M. Mendonça, and C. Seaman, "The practitioners' point of view on the concept of technical debt and its causes and consequences: A design for a global family of industrial surveys and its first results from Brazil," *Empirical Softw. Eng., Empirical Softw. Eng.*, vol. 25, no. 5, pp. 3216–3287, 2020. doi: 10.1007/s10664-020-09832-9.
4. N. Rios, M. G. Mendonça, and R. O. Spínola, "A tertiary study on technical debt: Types, management strategies, research trends, and base information for practitioners," *Inf. Softw. Technol.*, vol. 102, pp. 117–145, Oct. 2018. doi: 10.1016/j.infsof.2018.05.010.
5. Z. Li, P. Avgeriou, and P. Liang, "A systematic mapping study on technical debt and its management," *J. Syst. Softw.*, vol. 101, pp. 193–220, Mar. 2015. doi: 10.1016/j.jss.2014.12.027.
6. T. Besker, A. Martini, and J. Bosch, "Managing architectural technical debt: A unified model and systematic literature review," *J. Syst. Softw.*, vol. 135, pp. 1–16, Jan. 2018. doi: 10.1016/j.jss.2017.09.025.
7. T. Besker, H. Ghanbari, A. Martini, and J. Bosch, "The influence of Technical Debt on software developer morale," *J. Syst. Softw.*, vol. 167, p. 110,586, Sept. 2020. doi: 10.1016/j.jss.2020.110586.
8. P. Devanbu, T. Zimmermann, and C. Bird, "Belief & evidence in empirical software engineering," in *Proc. 38th Int. Conf. Softw. Eng. (ICSE)*, 2016, pp.108–119. doi: 10.1145/2884781.2884812.

ABOUT THE AUTHORS

NICOLLI RIOS is a postdoctoral researcher at PESC (Systems and Computer Engineering Program)/The Alberto Luiz Coimbra Institute for Graduate Studies and Research in Engineering, Federal University of Rio de Janeiro, Rio de Janeiro, 21941-972, Brazil. She is also a researcher with the Technical Debt Research Team. Her research interests include technical debt and empirical software engineering. Rios received her Ph.D. in computer science from the Federal University of Bahia. Contact her at nicolli@cos.ufrj.br.

SÁVIO FREIRE is a Ph.D. student in the Department of Computer Science at the Federal University of Bahia Salvador, Bahia, 40170-110, Brazil, and an assistant professor at the Federal Institute of Ceará, Morada Nova, Ceará, 62.940-000, Brazil. He is a researcher with the Technical Debt Research Team. His research interests include technical debt and empirical software engineering. Contact him at savio.freire@ifce.edu.br.

BORIS PÉREZ is an assistant professor at Francisco de Paula Stder. University's Department of Systems and Informatics, Cúcuta, 12E-96, Colombia. He is also a Ph.D. candidate at the Department of Systems and Computing Engineering, Universidad de Los Andes, Bogota, 111711, Colombia. His research interests include software architecture, model-driven architecture, and technical debt. Pérez received his M.Sc. in software engineering from the Universidad de Los Andes. Contact him at borisperezg@ufps.edu.co.

CAMILO CASTELLANOS is a Ph.D. candidate in the Department of Systems and Computing Engineering, Universidad de Los Andes, Bogota, 111711, Colombia. His research interests include software architecture, big data analytics, and model-driven engineering. Contact him at cc.castellanos87@uniandes.edu.co.

DARÍO CORREAL is an associate professor at Los Andes University, Bogota, 111711, Colombia. His research interests inlude software architecture, solutions architectures, service-oriented software engineering, and blockchain solutions. Further information about him can be found at https://profesores.virtual.uniandes.edu.co/dcorreal/es/inicio/. Contact him at dcorreal@uniandes.edu.co.

MANOEL MENDONÇA is a professor of computer science at the Federal University of Bahia (UFBA), Salvador, Ba, 40170-115, Brazil. He acted as the founding director of the Fraunhofer Center for Software and Systems Engineering at UFBA. His research interests include software visualization, empirical software engineering, and technical debt. Mendonça received his Ph.D. in computer science from the University of Maryland. Contact him at manoel.mendonca@ufba.br.

9. S. McConnell. "Technical debt," 10x Software Development Blog." Construx Conversations. 2008. https://www.construx.com/resources/whitepaper-managing-technical-debt/
10. W. Webber, A. Moffat, and J. Zobel, "A similarity measure for indefinite rankings," *ACM Trans. Off. Int. Syst.*, vol. 28, no. 4, 2010, Art. no. 20. doi: 10.1145/1852102.1852106.

ABOUT THE AUTHORS

DAVIDE FALESSI is an assistant professor (RTDb) at the University of Rome "Tor Vergata," Rome, 00133, Italy. His research interests include technical debt and machine learning to support software engineering tasks. Falessi received his Ph.D. from the University of Rome "Tor Vergata." Contact him at falessi@ing.uniroma2.it.

CAROLYN B. SEAMAN is a professor of information systems at the University of Maryland Baltimore County (UMBC), Baltimore, Maryland, 21250, USA. She is also the director of the Center for Women in Technology, also at UMBC. Her research interests focus on empirical studies of software engineering. Seaman received her Ph.D. in computer science from the University of Maryland, College Park. She is a Senior Member of IEEE. Further information about her can be found at https://userpages.umbc.edu/~cseaman/. Contact her at cseaman@umbc.edu.

CLEMENTE IZURIETA is an associate professor of computer science in the Gianforte School of Computing at Montana State University, Bozeman, Montana, 59717, USA. His research focuses on quality assurance, technical debt, and cybersecurity. He is active in many collaborative projects in which computing techniques enhance outcomes. He is also the chief technology officer of Authors A.I. (authors.ai). He is a Senior Member of IEEE. Contact him at clemente.izurieta@montana.edu.

RODRIGO SPÍNOLA is a professor of software engineering at Salvador University, Salvador, Bahia, 41770-235, Brazil, where he leads the Technical Debt Research Team. He is also a visiting professor at the State University of Bahia, Alagoinhas, Bahia, 48000-000, Brazil. His research interests include technical debt and empirical software engineering. Spínola received his Ph.D. in computer science and systems engineering from the Federal University of Rio de Janeiro. Contact him at rodrigo.spinola@unifacs.br.

11. N. Rios, R. O. Spínola, M. Mendonça, and C. Seaman, "Supporting analysis of technical debt causes and effects with cross-company probabilistic cause-effect diagrams," in *Proc. 2nd Int. Conf. Tech. Debt*, 2019, pp. 3–12. doi: 10.1109/TechDebt.2019.00009.
12. S. Freire et al., "Surveying software practitioners on technical debt payment practices and reasons for not paying off debt items," in *Proc. Eval. Assessment Softw. Eng.*, 2020, pp. 210–219. doi: 10.1145/3383219.3383241.