

Pitfalls and Solutions for Technical Debt Management in Agile Software Projects

S. Freire

Federal University of Bahia and Federal Institute of Ceará

N. Rios

Federal University of Rio de Janeiro

B. Pérez

University of Los Andes and Francisco de Paula S/der University

C. Castellanos

University of Los Andes

D. Correal

University of Los Andes

R. Ramač

University of Novi Sad

N. Taušan

INFORA Research Group

V. Mandić

University of Novi Sad

A. Pacheco

University of Costa Rica

G. López

University of Costa Rica

M. Mendonça

Federal University of Bahia

C. Izurieta

Montana State University and Idaho National Laboratories

D. Falessi

University of Rome “Tor Vergata”

C. Seaman

University of Maryland Baltimore Country

R. Spínola

Salvador University and State University of Bahia

Abstract—As technical debt (TD) management balances short-term and long-term goals, having information on the impediments, decision factors, enabling practices, and actions (IDEA) related to TD management can support software teams in improving their ability to effectively manage debt items. This article presents TD IDEA Diagrams for TD management in agile software projects. The diagrams are grounded in reports from 274 practitioners from six countries. They organize practices for TD prevention, monitoring, and repayment, as well as the impediments for applying them. By analyzing the diagrams, professionals can avoid the pitfalls, and increase their capacity, for better TD management.

■ **Technical debt (TD)** management encompasses the prevention, monitoring, and repayment of TD items [1], which are the result of intentional shortcuts or even mistakes taken in software development [2,8]. TD can sometimes bring a short-term benefit to the project, usually in terms of increased development speed or shortened time to market, but may have to be paid with interest later on [2]. There are specific practices for TD prevention, monitoring, and repayment, as well as practices that help software teams to better manage TD. There are also common reasons for not applying these practices, which include: (1) impediments that prevent teams who want to manage the debt from doing so; and (2) technical or administrative reasons that lead a team to decide against managing TD effectively [3].

Identifying these practices and practice avoidance reasons (PARs) aids in choosing appropriate strategies to control TD. Moreover, understanding how they interact in combination allows a comprehensive view of the TD management landscape. In this article, we present IDEA Diagrams (**I**mpediments, **D**ecision factors, **E**nabling practices, and **A**ctions) to help frame TD management. Loosely inspired by SWOT (strengths, weaknesses, opportunities, and threats) analysis [5], the IDEA Diagrams organizes TD management practices and PARs into quadrants. This paper presents IDEA diagrams for TD prevention, monitoring, and repayment in agile projects. To populate the diagrams, we use the practices and PARs reported by 274 practitioners from the agile software industry who responded to the InshighTD survey [6].

This article supports practitioners in:

- Identifying common practices used for preventing, monitoring, and repaying debt items;
- Starting or improving TD management initiatives, presenting the PARs that curb the improvement of the team's ability to manage debt;
- Understanding impediments to TD management practices that can be addressed in TD management initiatives.

Overall, identifying and understanding key issues (impediments and decision factors) and capabilities

(actions and enabling practices) for TD management support agile teams in balancing the long-term and short-term goals of a development project, contributing to the effectiveness of their TD management initiative. See [9,10] to learn more about TD and agile software development.

HOW WE BUILT THE DIAGRAMS

Our work is part of the InshighTD Project, which is a globally distributed family of industrial surveys on TD causes, effects, and management, involving researchers from multiple countries [6]. The survey begins with characterization questions, then proceeds to ask the participants to define TD and to describe a particular example of TD. Only participants with valid responses to those questions are considered for data analysis [6]. Then, for each of the investigated TD management activities (prevention, monitoring, and repayment) it asks if the participant performed the activity (yes/no question). If “no”, it asks why. If “yes”, it asks how. Thus, for example, if a participant indicates that it is possible to prevent the TD item (yes), practices for TD prevention are elicited; otherwise (no), the participant provides reasons for not preventing the debt. A detailed description of the survey instrument, as well as its planning and data analysis procedures are presented in [6]. Several results from the InshighTD project (<http://www.td-survey.com/publication-map/>) have been previously published, the novelty of this paper is twofold: 1) it addresses TD management practices (particularly TD monitoring) in agile projects for the first time, and 2) it describes for the first time the IDEA diagrams.

Our findings are based on the responses (available at <https://bit.ly/3w5hfCZ>) from 274 practitioners from six countries all of whom indicated, in one of the characterization questions, working with agile processes. We used open coding [3,4,7] to extract the practices and PARs from the surveys' textual responses. Two researchers independently coded the entire set of responses from Brazil, followed by discussions aimed at reaching consensus. The InshighTD teams in Chile, Colombia, Costa Rica, Serbia, and the US then used the list of codified practices and PARs to standardize the nomenclature

found in their results. Afterwards, the Brazilian team reviewed and merged these results to provide the final list that forms the basis of the IDEA analysis (see **Sidebar 1**).

(Sidebar 1) Threats to the Study Validity

The main threat to the validity of the study is related to the coding of practices and PARs, which is a subjective task and could lead to biased results. Our analysis process, involving iteration, review, and multiple coders, mitigates this. We also reduced external validity threats by focusing the study on practitioners from agile software industry with diverse experience and work environments. The InsignTD questionnaire and the study planning are subject to some internal validity threats. In order to face these, we performed three internal validations by researchers from the project who didn't work in its planning and an external independent researcher. Lastly, we also run a pilot study to check process, instruments, and procedures consistency [6].

non-prevention, 22 for non-monitoring, and 19 for non-repayment of TD, our PARs. We identified two types of practices: actions and practices for increasing a team's ability to manage TD. The former is used to prevent, monitor, or repay debt items, such as *following the project planning*, *creating a TD item backlog*, and *code refactoring*, respectively. The latter improves the capacity of the team to perform those actions, for example, *training* and *use of tools*. We categorized the PARs into decision factors and impediments. A decision factor indicates that not managing the debt is a choice, for example, *lack of interest* and *focusing on short term goals*. Alternatively, impediments prevent a team who wants to manage the debt from doing so, such as *short deadline* and *cost*.

THE IDEA DIAGRAMS FOR TD PREVENTION, MONITORING, AND REPAYMENT

Inspired by SWOT analysis, which supports the definition of strategies to achieve organizational goals, organizing the strengths, weaknesses, opportunities, and threats into a matrix [5], the IDEA diagrams organize issues (decision factors and impediments) and capabilities (actions and enabling practices) into four quadrants. The scope of IDEA diagrams, however, is not organizational planning. They are aimed at supporting software teams concerned with TD management. They work as a communication tool, representing, in a simplified way, concerns that practitioners should have when improving the management of TD.

Figure 1 presents demographic information. Although it is not possible to guarantee that participants represent all practitioners working with agile processes in the software industry in those countries, the sample encompasses a broad and diverse set of practitioners, which help strengthen the analytical generalizability of the study.

In total, we identified 73 prevention, 25 monitoring, and 17 repayment practices, as well as 15 reasons for

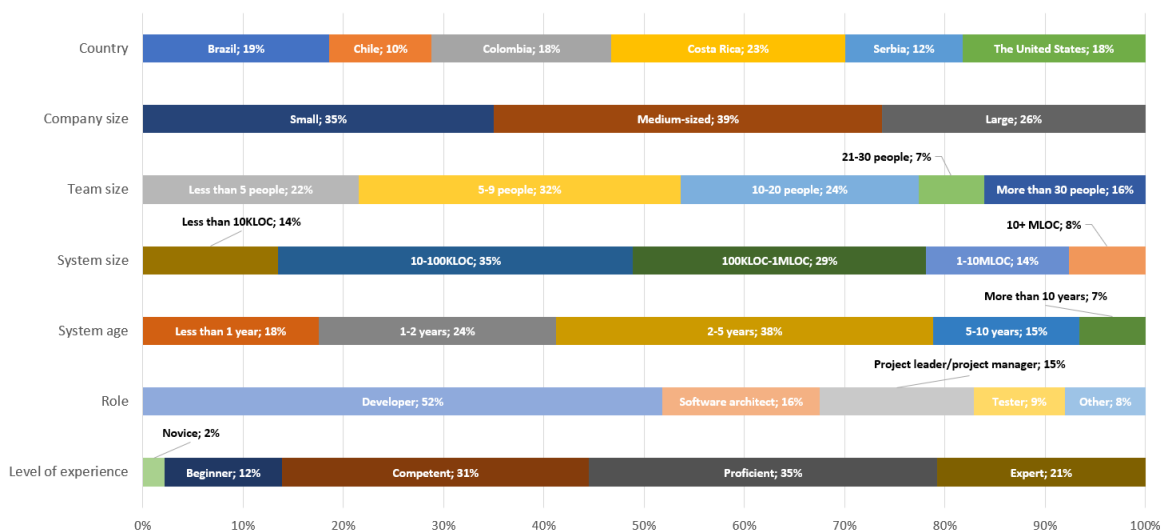


Figure 1. Overview of participants' characterization.

IDEA analysis can reveal capabilities or issues that the project team can improve, maintain, or reduce to manage debt items. To this end, we created the TD

IDEA diagrams for TD prevention, monitoring, and repayment presented in Figure 2. They organize the TD management practices and PARs into the matrix's

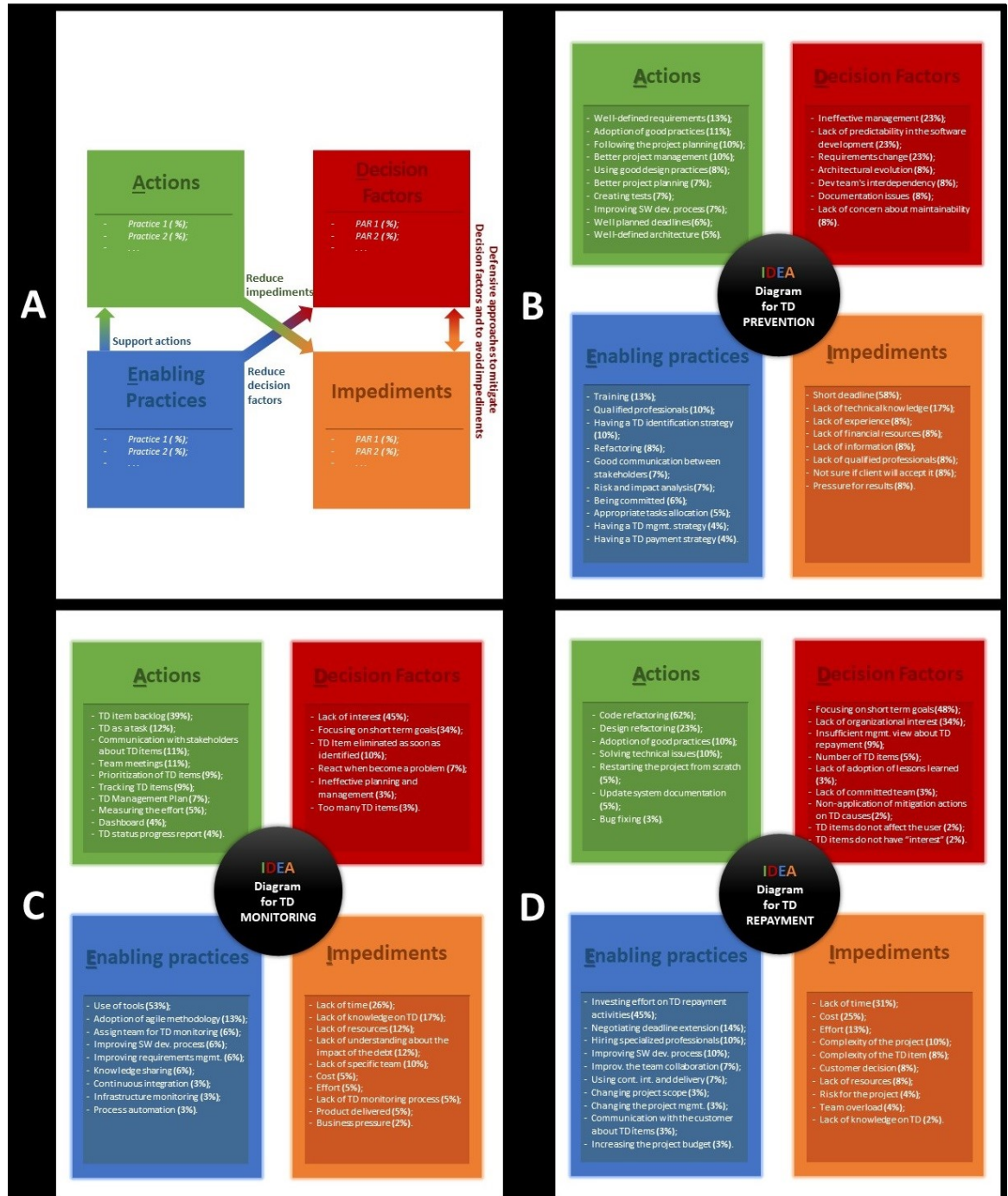


Figure 2. TD IDEA conceptual model (A) and TD IDEA diagrams for TD prevention (B), monitoring (C), and repayment (D) in agile processes.

quadrants following the type of each TD management variable. TD preventive, monitoring, and repayment actions indicate a team's capability to deal with debt items, while enabling practices can increase team's capabilities for improving TD management. Impediments represent factors out of the control of the team. Finally, the decision factors reveal the team's points of view that lead them to not prevent, monitor, or repay debt items.

The practices and PARs are presented in their corresponding quadrants ordered by how frequently they were reported in the survey (as %). For example, *short deadline (58%)* means that 58% of agile practitioners who reported an impediment for TD prevention (see **Figure 2B**) experienced it in their projects. These percentages do not indicate if a practice or a PAR is critical. In order to make the diagrams concise, the work considers only the 10 most frequently cited practices and PARs. A detailed analysis of them by the variables presented in Fig. 1 is available at <https://bit.ly/3w5hfCZ>.

USING THE DIAGRAMS

IDEA analysis can assist with the definition of TD management strategies by analyzing one or two quadrants of the matrix at a time. When looking at isolated quadrants, on the left side of each diagram, the team can see which practices can be applied to prevent, monitor, or repay debt items (actions) and also which practices help enable the team to better manage debt items (enabling practices). For example, **Figures 2B, 2C, and 2D** show that *well defined requirements*, *TD item backlog*, and *code refactoring* are the most commonly used practices to prevent, monitor, and repay debt items, respectively. *Training*, *use of tools*, and *investing effort on TD repayment* are the most common practices used to create a development environment conducive to debt prevention, monitoring, and repayment, respectively. This constitutes valuable information for software teams initiating TD management. Teams who already have an established management process can use this as a benchmark to analyze and improve their current practices.

On the right side of the diagram, teams can see the decision factors that lead to not managing debt items and the impediments that restrict TD management. Surmounting these PARs can be decisive to successfully manage debt items. For example, the most commonly found impediments for TD prevention, monitoring, and repayment are, respectively, *short*

deadlines, *lack of time*, and again *lack of time*. This is a strong indication that managing development time is essential to putting a TD management strategy in place. The PARs that negatively affect decisions to manage debt items include *ineffective management*, *lack of interest*, and *focus only on short term goals*. Changing the team's mindset on the importance of managing TD is definitely a key issue there.

Examining relationships between quadrants can also be useful. For example, agile teams can reduce weak areas by considering the enabling practices and decision factors quadrants. For instance, if the team wants to reduce its *lack of predictability in the software development* (weakness) in order to better prevent TD, **Figure 2B** suggests that adopting *risk and impact analysis* and *refactoring* (opportunities) could help by minimizing the chances of unexpected events during software development, thus boosting predictability and TD prevention.

Analyzing the actions and impediments quadrants can help a development team reduce the impediments to TD prevention. For example, by examining **Figure 2B**, the team could recognize *short deadlines* as one of its impediments to TD prevention, and also see that investing in *following the project planning*, and *well-planned deadlines* (actions), could help overcome this impediment by ensuring that deadlines are reasonable and designed to incorporate TD prevention actions from the beginning.

As another example, consider a team trying to understand why they are not monitoring TD effectively. If the team examines the decision factors and impediments quadrants of **Figure 2C**, they could recognize *lack of interest* as a decision made by their team, meaning that few managers and developers seem to be interested in monitoring TD at all. This is a problem, obviously, and some clues to addressing it could be found in the impediment quadrant, specifically *lack of knowledge on TD* and *lack of understanding about the impact of the debt*. The team might then conclude that addressing these impediments, possibly through education and collection of data about the evolution of the product, could address the *lack of interest* and lead to better monitoring of TD.

Finally, examining the actions and enabling practices quadrants provide teams with a way to boost their TD management ability by suggesting other practices that could be implemented. For example, **Figure 2D** indicates that *code refactoring*, *design*

refactoring, and *solving technical issues* are repayment activities (actions). If a team recognizes these actions in their own development process but wants to build on them, the IDEA diagram suggests enabling practices for activities like *investing effort on TD repayment activities* and *negotiating deadline extension*, which would create more resources for the repayment activities to be even more effective, and would also increase the chances of continuously investing in TD repayment actions.

CONCLUSION

Some of the insights given by the IDEA diagrams may seem obvious to experienced managers. However, the diagrams provide a framework to ensure a holistic analysis for TD prevention, monitoring, and payment. For example, in the agile context:

- Repayment and monitoring actions are more commonly related to technical and managerial activities, respectively, while preventive actions are usually related to both;
- Training, using tools, and investing effort on TD repayment increase the team's capability to manage TD;
- Ineffective management, lack of interest, and focusing on short-term goals are the primary decision factors for do not manage TD items;
- Lack of time and short deadlines are the main impediments to perform TD management.

For an agile team who wants to start managing TD, the ranked lists of practices and PARs organized in each of the TD IDEA diagrams provide valuable guidance on what to employ (practices) or curb (PARs) based on experience from other development teams. If a team already has experience in managing TD, it can identify other commonly used practices or other PARs faced, and can also identify enabling activities (enabling practices) that will improve the team's ability to manage TD. In other words, teams can create their own TD IDEA diagrams. This exercise is beneficial in and of itself, but is also useful in comparing a team owned diagram to those of others, providing learning opportunities on TD management among teams or squads. Thus, we stimulate the readers to go further, creating TD IDEA diagrams for their own context and comparing them to the ones presented in this article.

REFERENCES

1. 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," *Information and Software Technology*, Vol.102, pp.117-145, 2018.
2. P. Avgeriou, P. Kruchten, R. L. Nord, I. Ozkaya and C. Seaman, "Reducing Friction in Software Development," in *IEEE Software*, vol. 33, no. 1, pp. 66-73, Jan.-Feb. 2016, doi: 10.1109/MS.2016.13.
3. S. Freire, N. Rios, B. Gutierrez, D. Torres, M. Mendonça, C. Izurieta, C. Seaman, and R. Spínola, "Surveying Software Practitioners on Technical Debt Payment Practices and Reasons for not Paying off Debt Items". In *EASE*, pp.210–219, 2020.
4. Seaman C (1999) Qualitative methods in empirical studies of software engineering. *IEEE Transactions on Software Engineering*, 25(4):557-572.
5. H. Shahir, S. Daneshpajouh, and R. Ramsin, "Improvement Strategies for Agile Processes: A SWOT Analysis Approach," *Proc. Sixth International Conference on Software Engineering Research, Management and Applications*, pp. 221-228, 2008.
6. 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 Software Engineering*, Vol.25, pp.3216–3287, 2020.
7. S. Freire, N. Rios, M. Mendonça, D. Falessi, C. Seaman, C. Izurieta, and R.O. Spínola, "Actions and impediments for technical debt prevention: results from a global family of industrial surveys". In *ACM SAC*, pp.1548-1555, 2020.
8. D. Falessi and R. Kazman, "Worst Smells and Their Worst Reasons," In *TechDebt*, pp.45-54, 2021.
9. N. Rios *et al.*, "On the Relationship Between Technical Debt Management and Process Models," in *IEEE Software*, 2021, doi: 10.1109/MS.2021.3058652.
10. W. N. Behutiye, P. Rodríguez, M. Oivo, and A. Tosun, "Analyzing the concept of technical debt in the context of agile software development: A systematic literature review," *Information and Software Technology*, Vol.82, pp.139-158, 2017.



SÁVIO FREIRE is a PhD student at the Department of Computer Science at the Federal University of Bahia, Salvador, 40170-110, Bahia, Brazil, and an assistant professor at the Federal Institute of Ceará, Morada Nova, Ceará, 62.940-000, Brazil. He is a researcher at the Technical Debt Research Team (www.tdresearchteam.com) at the Salvador University. His research interest encompasses technical debt and empirical software engineering. Contact him at savio.freire@ifce.edu.br.



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



NICOLLI RIOS is a postdoctoral researcher at PESC (Systems and Computer Engineering Program)/COPPE (The Alberto Luiz Coimbra Institute for Graduate Studies and Research in Engineering) at Federal University of Rio de Janeiro, Rio de Janeiro, 21941-972, Brazil. She is also a researcher at the Technical Debt Research Team (www.tdresearchteam.com). She holds a PhD in Computer Science from the Federal University of Bahia. Her main research interests are technical debt and empirical software engineering. Contact her at nicolli@cos.ufrj.br.



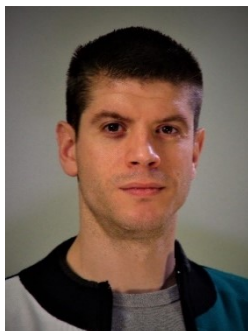
DARÍO CORREAL, PhD, is associate professor at Los Andes University Bogota, 111711, Colombia. His main research topics are software architecture, solutions architectures, service-oriented software engineering and blockchain solutions. More can be found at <https://profesores.virtual.uniandes.edu.co/dcorreal/es/inicio/>. Contact him at dcorreal@uniandes.edu.co.



BORIS PÉREZ is an assistant professor in Francisco de Paula Stder. University's Department of Systems and Informatics in Cúcuta, 12E-96, Colombia. Also, he is PhD 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. Boris received his MSc in software engineering from Universidad de Los Andes. Contact him at borisperezg@ufps.edu.co.

ROBERT RAMAČ is a teaching assistant of Software Engineering at the University of Novi Sad, Faculty of Technical Sciences, 21000 Novi Sad, Serbia, and a software developer at TIAC ltd. Currently he is enrolled as a PhD student at the University of Novi Sad, at the

Faculty of Technical Sciences. He received two M.Sc. degrees at the University of Novi Sad, master of Information technologies and master of Engineering management. His areas of interest are: empirical software engineering, software development, technical debt, the improvement of the software development process. Contact him at ramac.robert@uns.ac.rs.



NEBOJŠA TAUŠAN is a software engineer at INFORA Research Group, Adolfa Singera 12/A, 24000 Subotica, Serbia. His research interests include empirical software engineering, software architectures, DSLs, and technical debt. Nebojša holds a PhD degree in Information Processing Science and Software Engineering from University of Oulu, Finland, MSc and BSc degrees in Information Systems from Faculty of Economics Subotica, University of Novi Sad, Serbia. Contact him at nebojsa.tausan@infora.rs.



VLADIMIR MANDIĆ is an assistant professor of software engineering at University of Novi Sad, Faculty of Technical Sciences, 21000 Novi Sad, Serbia. His research interests include empirical software engineering, software process improvement, software quality, and technical debt. He holds a PhD degree in Information Processing Science and Software Engineering from University of Oulu, Finland, MSc and BSc degrees in Computer Science from University of Novi Sad, Serbia. Contact him at vladman@uns.ac.rs.



ALEXIA PACHECO (alexia.pacheco@ucr.ac.cr) is a Ph.D. candidate at the graduate program in Computer Science at the University of Costa Rica, Costa Rica. Her research interests include technical debt, software analytics, and empirical software engineering. She holds an MS in Applied Mathematics and a Degree in Computer Science from the University of Costa Rica.



GUSTAVO LOPEZ is a professor of software engineering and human-computer interaction at the University of Costa Rica (San Pedro, San Jose, Costa Rica). He holds a Ph.D. in Computer Science from the same university. His research interests include agile software development, software process improvement, technical debt, user experience, and usability. Contact him at gustavo.lopez_h@ucr.ac.cr.



MANOEL MENDONÇA is a professor of computer science at the Federal University of Bahia (UFBA), Salvador, Ba, 40170-115, Brazil. He holds a PhD in Computer Science from the University of Maryland. His main research interests include software visualization, empirical software engineering, and technical debt. Contact him at manoel.mendonca@ufba.br.



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 where 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.



DAVIDE FALESSI is an assistant professor (RTDb) at the University of Rome “Tor Vergata,” Rome, 00133, Italy. He got his bachelor, master, and PhD degrees from that same university. His research interests include technical debt and machine learning to support software engineering tasks. You can contact him at falessi@ing.uniroma2.it.



CAROLYN 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 consists mainly of empirical studies of software engineering, with particular emphases on maintenance, organizational structure, communication, measurement, and technical debt. She also investigates qualitative research methods in software engineering, as well as computing pedagogy. She holds a PhD in Computer Science from the University of Maryland, College Park, a MS from Georgia Tech, and a BA from the College of Wooster (Ohio). More can be found at <https://userpages.umbc.edu/~cseaman/>. Contact her at cseaman@umbc.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 (www.tdresearchteam.com). 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. He holds a PhD and MS in Computer Science and Systems Engineering from the Federal University of Rio de Janeiro. Contact him at rodrigo.spinola@unifacs.br.