Organizing the State of Practice on Technical Debt Prevention, Monitoring, and Payment in Software Projects

Sávio Freire[1], Rodrigo Spínola[2] and Manoel Mendonça[3]

[1]Teaching Department – Federal Institute of Ceará (IFCE)
Morada Nova – CE – Brazil

[2]Computer Science Department – Virginia Commonwealth University (VCU)
Richmond, VA – United States

[3]Institute of Computing – Federal University of Bahia (UFBA)
Salvador – BA – Brazil

savio.freire@ifce.edu.br, spinolaro@vcu.edu, manoel.mendonca@ufba.br

Abstract. *Knowing technical debt (TD) prevention, monitoring, and payment practices can help development teams choose the best practices for their projects. Identifying the practice avoidance reasons that lead to non-prevention, non-monitoring, and non-payment of TD can support teams in increasing their ability to manage TD items. This Ph.D. dissertation aims to investigate the state of practice in managing TD items in software projects through the continuous and independent replicating of a family of globally distributed surveys named InsighTD. We conducted a literature review on TD prevention, monitoring, and payment and then analyzed data collected by six InsighTD replication teams. The results yielded three types of artifacts: (1) an updated TD conceptual model, (2) a set of TD management maps, and (3) the IDEA diagrams listing TD Impediments, Decision factors, Enabling practices, and Actions. Lastly, we assessed these artifacts through empirical studies in academia and industry.*

1. Introduction

Technical debt (TD) describes the effect of immature artifacts on the software development process, bringing benefits to projects in the short term but which might have to be adjusted with interest later. The benefits are higher productivity and lower costs. At the same time, the "interest" is associated with unexpected delays in system evolution activities and the difficulty in achieving the quality criteria defined for the project [Zazworka et al. 2013].

Several studies [Li et al. 2015, Rios et al. 2018] have addressed identifying TD items and strategies for their management. As not all TD items identified in the project may not influence the evolution of the software [Brown et al. 2010], a project team can balance short-term and long-term goals through TD management activities. They can support decision-making on the need and the best time to eliminate a debt item [Guo et al. 2016].

TD management comprises the identification, measurement, prioritization, prevention, monitoring, documentation, communication, visualization, time-to-market analysis, scenario analysis, and payment of debt items [Li et al. 2015, Rios et al. 2018].

Among those, TD prevention, monitoring, and payment are critical. **TD prevention** enables software development teams to prevent TD items from occurring. **TD monitoring** seeks to support software teams in observing changes in the cost and benefit of debt items that have not yet been eliminated during the project. **TD payment** activity supports decision-making about the most appropriate time to pay debt items and the choice and application of practices that should be used to pay TD items.

Many authors have researched the practices used by software professionals to prevent, monitor, and pay TD items [Li et al. 2015, Ernst et al. 2015, Martini et al. 2018, Apa et al. 2020, Aragão et al. 2022]. However, the results are limited as most studies address only one type of debt and focus on specific case studies. [Ernst et al. 2015], [Martini et al. 2018], and [Apa et al. 2020] considered a sizable number of participants but considered a limited set of software development contexts.

There is a knowledge gap regarding the practices used to prevent, monitor, and pay off TD items, and the reasons used to justify the non-prevention, non-monitoring, and non-payment of TD items. Herein, these reasons will be called *practice avoidance reasons (PARs)*. Identifying the practices used by software practitioners to manage the debt and the PARs that lead them not to manage debt is fundamental to guiding new research related to TD. Knowing the management practices can help software practitioners identify new practices they have not yet used. Knowing the PARs can aid teams in understating which aspects need to be improved to prevent, monitor, and pay off TD items.

This Ph.D. dissertation [Freire 2023] aims **to investigate, through continuous and independent replication of a globally distributed family of surveys, the state of practice on TD prevention, monitoring, and payment in software projects**. To this end, we defined the following research questions (RQs): **(RQ1)** How could software development teams avoid TD items on their projects?, **(RQ2)** How have software development teams monitored TD items on their projects?, **(RQ3)** How have software development teams paid off TD items on their projects?, and **(RQ4)** How to organize the body of knowledge composed of prevention, monitoring, and payment practices - and PAR for TD non-prevention, non-monitoring, and non-payment - to support TD management?

To achieve the goal and answer the RQs, we defined four specific goals (SGs): **(SG1)** To investigate the current state of TD research by identifying studies that addressed TD prevention, monitoring, and payment; **(SG2)** To investigate the state of the practice on TD prevention, monitoring, and payment; **(SG3)** To organize the body of knowledge composed of practices and PARs identified in SG2; and **(SG4)** To assess the body of knowledge organized in SG3.

Besides this introductory section, this paper is composed of four sections. Section 2 presents the research method used in the dissertation. Section 3 presents the TD prevention's, monitoring's, and payment's state of practice. In Section 4, we present the artifacts we defined by the body of knowledge and the empirical studies used to assess them. Lastly, Section 5 presents the final considerations of this work.

2. Research Method

The activities defined and performed in this Ph.D. dissertation are based on the Experimental Software Engineering paradigm [Basili 1993], which aims to assist in better eval-

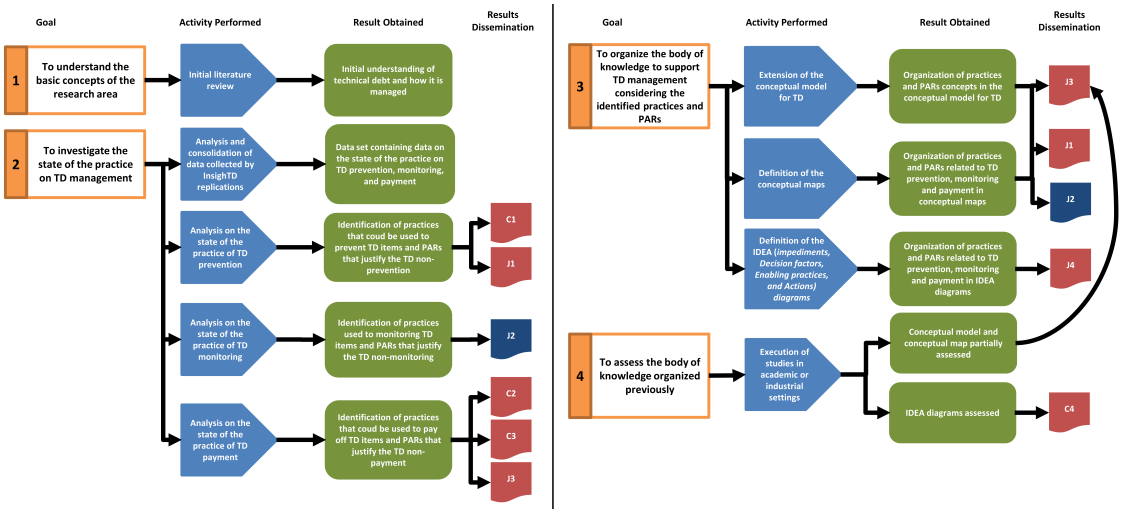

Figure 1. Ph.D. dissertation methodology and results

uating, predicting, understanding, controlling, and improving software engineering practices [Basili et al. 1986].

This Ph.D. dissertation is based on a family of surveys named the InsighTD Project [Rios et al. 2020]. This project aims to examine the state of software engineering TD practice, revealing its causes, effects, and management practices. The InsighTD survey was designed to allow its replication in different countries. It was defined in the context of a Ph.D. dissertation [Rios Alves 2020] aiming at investigating the causes and effects of TD. The project also addresses TD management (prevention, monitoring, and payment) by asking the participants whether these activities have been performed and giving details about how these activities were performed or why they were not. Our Ph.D. dissertation uses the data collected from the InsighTD Project to investigate the state of the practice of TD prevention, monitoring, and payment.

Figure 1 presents the methodological steps we followed to achieve the work's goal. Each step comprises four elements: goal, activity performed, the result obtained, and result dissemination. The goals are represented by white rectangles and their activities by blue pentagons. The obtained results are represented by rounded green rectangles.

Initially, we conducted a literature review on TD, aiming to understand its concept and state of the art on TD, meeting SG1. Next, we analyzed a data set composed of answers collected by the Brazilian and North American InsighTD replication teams. Also, we consolidated the analysis with the results of Chilean, Colombian, Costa Rican, and Serbian InsighTD replication teams. In total, our data set comprises 653 responses from software professionals from these six countries. Then, we analyzed the data from the InsighTD survey's questions on TD prevention, monitoring, and payment to identify practices and PARs, achieving SG2.

To accomplish SG3, we extended the conceptual model for TD [Rios et al. 2018], defined three conceptual maps that summarizes the practices and PAR we found in the state of practice, and defined the impediments, decision factors, enabling practices, and actions (IDEA) diagrams, explaining their structure and how to use them for supporting TD management activities. Lastly, we planned and conducted three studies to assess the conceptual model, conceptual maps, and IDEA diagrams, meeting SG4. By conducting a

follow-up survey with InsighTD participants, we partially evaluated the conceptual model and the TD payment map. By completing a TAM study with undergraduate students, we characterized the IDEA diagrams concerning ease of use, usefulness, and potential future use. By conducting an interview-based study with experienced software practitioners, we characterized the experts' perceptions of the diagrams concerning their support for TD management activities.

The last column of Figure 1 shows the disseminated results. We used an icon to describe them, that is, articles that have been published (red icon) or that are just accepted (blue icon). We will discuss on the publications in Section 5.2.

3. Results

From the InsighTD survey, we analyzed 653 answers, including data from Brazil (107 answers), Chile (89), Colombia (134), Costa Rica (145), Serbia (79), and the United States (99). This data helps us understand the state of the practice of TD prevention (Section 3.1), monitoring (Section 3.2), and payment (Section 3.3).

Figure 2 summarizes the participants' characterization by country, company and team size, system size and age, role, level of experience, and process model. Overall, the collected data includes a wide variety of projects from the software development industry in Brazil, Chile, Colombia, Costa Rica, Serbia, and the United States, including projects of different ages, sizes, team sizes, process models, and composed of several participant roles and levels of experience from organizations of different sizes.

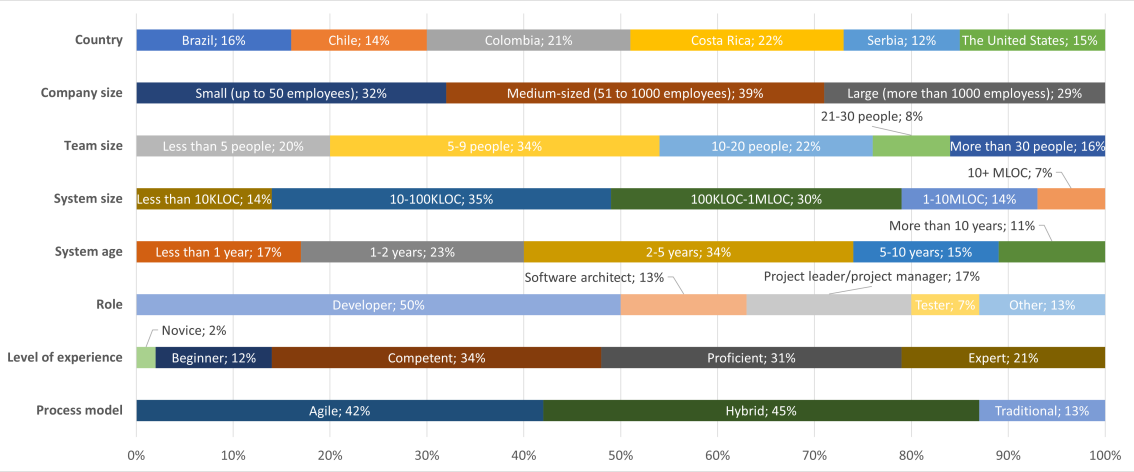

Figure 2. Summary of InsighTD project participant's characterization

3.1. TD Prevention's State of Practice (RQ1)

We identified 89 prevention-related practices reported by software practitioners. The practices *well-defined requirements*, *adoption of good programming practices*, and *better project management* were the most cited, indicating that improvements applied to software requirements, code, and management are necessary to increase the team's ability to avoid debt items. Also, we found that the practices can directly prevent debt items (prevention actions) or support TD prevention initiatives (enabling practices). We grouped all practices into seven categories related to software development concerns. The categories

planning & management and *methodology* stand out, encompassing the greatest number of prevention-related practices.

Regarding PARs for TD non-prevention, we found 23 of them. *Short deadline*, *ineffective management*, *lack of predictability in the software development*, and *requirements change* were the most cited. PARs can be either an external factor outside the control of the team (an impediment) or a decision of the team itself (a decision factor), revealing that the non-prevention of TD can be a decision of the team or some external factor affecting the project, such as the organization or the customer, among others. PARs were grouped into seven categories. The categories *planning & management*, *development issues*, and *methodology* have the largest number of PARs, revealing that decisions taken in software development and methodology and at the managerial level are decisive for TD prevention. We also found that the *planning & management* category encompasses the PARs that can explain the non-prevention of almost all investigated types of debt.

3.2. TD monitoring's State of Practice (RQ2)

We identified 46 monitoring-related practices for monitoring TD items. Of these practices, *TD item backlog*, *use of tools*, *team meetings*, and *improving software development process* are the most used. Practitioners have used the identified practices to monitor, prevent, identify, and pay off TD items, and enable TD monitoring initiatives. It reveals that efforts to monitor TD items should consider the integration among these TD management activities. Also, we grouped the practices into seven categories. *Planning & management* and *methodology* encompass many of the practices, playing a leading role in TD monitoring. Lastly, the ten most cited practices are performed for monitoring the analyzed 14 types of debt. Practices from *planning & management* are used for monitoring almost all analyzed types, indicating that it would be a viable choice to start TD monitoring initiatives by them.

We identified 35 PARS used for explaining the non-monitoring of TD items. *Lack of interest*, *focusing on short term goals*, *lack of time*, and *lack of knowledge on TD* were the most commonly cited PARs. They are classified into two types: decision factors and impediments. Impediments are more commonly considered for justifying the non-monitoring of TD items than decision factors. Monitoring TD is not just a matter of will but of mitigating the restrictions that curb the adoption of monitoring practices. We group the PARs into eight categories. *Planning & management* and *organizational* concentrate the greatest number of PARs, revealing that decisions taken at organizational or managerial levels are decisive for monitoring TD items. We have also investigated the relation between PAR and types of debt. The ten most cited PAR have been considered to justify the non-monitoring of all types of debt. Particularly, PARs from *planning & management* and *organizational* have been used to explain most of the analyzed types.

3.3. TD Payment's State of Practice (RQ3)

We found a set of 32 practices related to the payment of TD, which fall into four broad types: practices that directly result in debt item payment, practices that help create a favorable scenario for future debt payment, TD prevention practices, and TD prioritization practices. *Code refactoring*, *investing effort on TD payment activities*, and *design refactoring* are the most cited practices. We also found that the identified practices are more

commonly concentrated in *methodological* issues of software development, and practitioners have used them for paying off several types of debt: architecture, build, code, defect, design, documentation, infrastructure, people, requirements, service, and test. This is an indication that the process followed by software practitioners plays a central role for TD payment. Process improvements seem to be a good starting point for TD payment initiatives.

Regarding PARs for TD non-payment, we identified 27 PARs of them. Among them, *focusing on short term goals*, *lack of organizational interest*, and *lack of time* are the most cited PARs, revealing that managerial decisions are quite decisive. We identified that these PARs can be either a decision taken by the team to intentionally not pay off debt or an impediment that hinders the payment of debt items regardless of the practitioners' intentions. *Impediments* are slightly more commonly faced ($\sim$65%) than *decision factors*. The PARs are more used for justifying the non-payment of design, test, code, architecture, documentation, and requirements debt items. Also, we found more managerial PARs for TD non-payment than technical ones, indicating that the management view is decisive for the non-payment of TD.

4. Organizing the Body of Knowledge on TD Management (RQ4)

This section organizes the state of practice of TD prevention, monitoring, and payment activities using different views to consolidate the set of information on TD management.

4.1. TD Conceptual Model and TD Management Conceptual Maps

Initially, we extended the conceptual model for TD [Rios et al. 2018], including the concepts of TD prevention, monitoring, and payment from the state of practice. Figure 3 shows the extended conceptual model. Our extension includes all the organized empirical evidence in this work related to the concept of TD prevention, monitoring, and payment (classes in green). These classes are associated with two lists. One is formed by the PARs (*PracticeAvoidanceReason* class), and the other represents the related practices (*PreventionRelatedPractice*, *MonitoringRelatedPractice*, and *PaymentRelatedPractice* classes). More details on the extension is available in [Freire 2023].

We also organized the set of practices and PARs in conceptual maps. For this, we were inspired by evidence briefings used to disseminate research findings to practitioners [Cartaxo et al. 2016]. Instead of using a one-page document to transfer knowledge acquired from empirical studies, we defined a set of maps composed of categories and their practices and PARs, along with their types and nature. We depict a map for each TD management activity (prevention, monitoring, and payment).

Figure 4 presents the TD payment map. The maps organize practices and PARs grouped by category. The rectangles with rounded edges group the entire set of practices and PARs. Rectangles with dashed lines represent the categories of practices and PARs. In each category, the map shows the percentage associated with the category and its practices or PARs. The higher the percentage, the greater the number of citations received.

Small rectangles indicate the nature of a practice or PAR, with black rectangles indicating a technical nature, while white rectangles denote managerially. For example, the *development issues* practice category has three managerial (*changing project scope*, *restarting the project from scratch*, and *system retirement*) and three technical practices

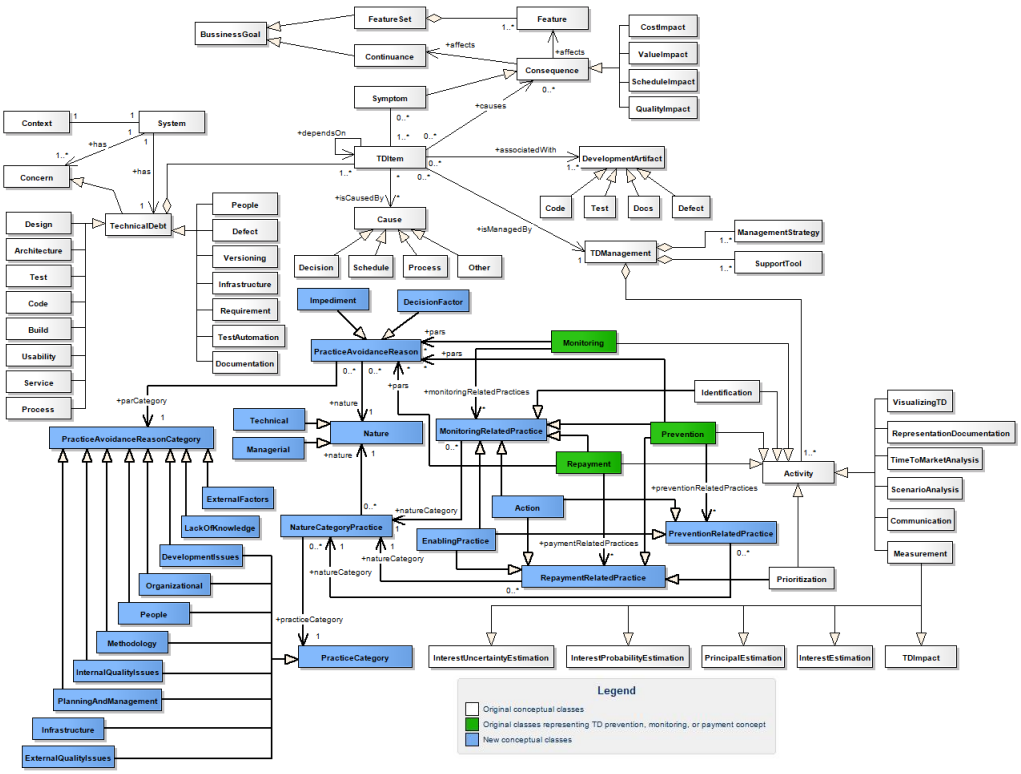

Figure 3. Extended conceptual model for technical debt

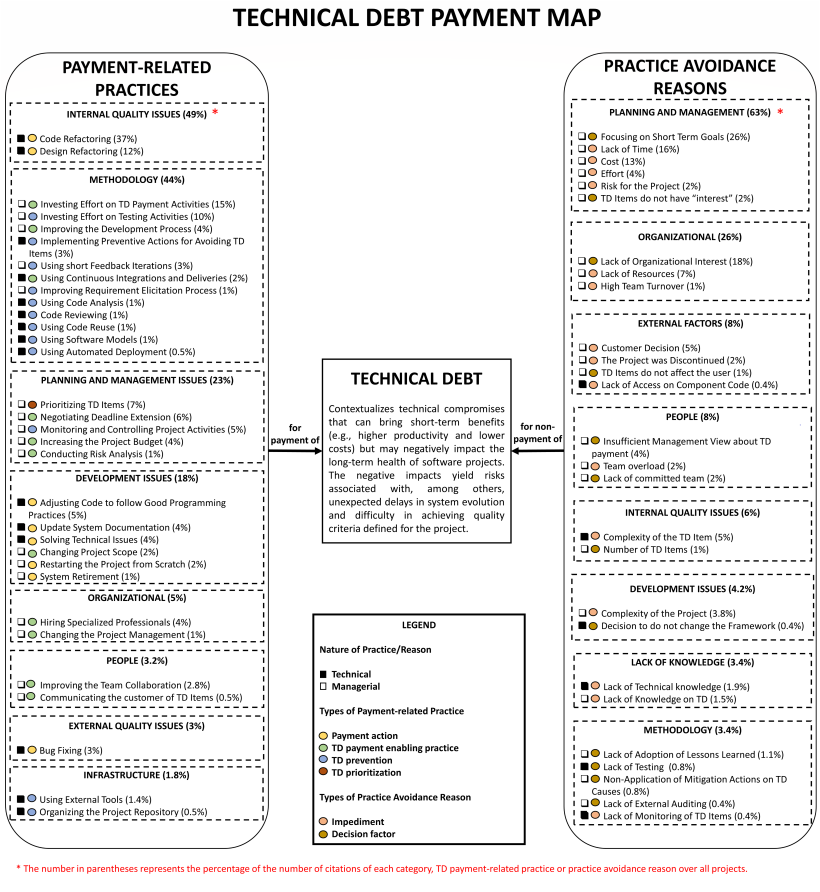

Figure 4. Technical debt payment map

(*adjusting code to follow good programming practices*, *update system documentation*, and *solving technical issues*). Small circles represent the type of practice or PAR. The *organizational* category has one brown circle (*lack of organizational interest*), representing a PAR of the *decision factor* type, and two orange circles (*lack of resources* and *high team turnover*) of the *impediment* type.

In the Ph.D. dissertation [Freire 2023], we present the other conceptual maps, give more details on them, and explain how to use them.

4.1.1. Assessing the Model and the Map

We conducted a follow-up survey with eight InsighTD participants ($\sim$30% response rate) to investigate the perception of software practitioners on the accuracy and completeness of the proposed TD conceptual model and map. However, for reasons of cost and focus, we only considered (i) a version of the model containing only the extension for TD payment concepts and (ii) the TD payment conceptual map, as shown in Figure 4.

The follow-up survey was composed of eleven questions, divided into two sections. In the first, the participants provided their perception of the TD conceptual model, and the latter captured the participants' perceptions of the TD payment map. In both sections, we provided an explanatory video to support the participants to understand the model and the map.

Evaluation results from the conceptual model to TD payment indicate that it is well organized, accurate and complete, as well as provides valuable information to define strategies for TD payment. Regarding the TD payment map, the assessment reveals that the map seems useful as a support tool in TD payment activities, but it must be adapted according to practitioners' context. Unfortunately, we cannot extrapolate these results to the full version of the model or to TD prevention and monitoring maps, requiring new investigations on those artifacts.

4.2. IDEA Diagrams

IDEA diagrams organize issues — decision factors and impediments — and capabilities — actions and enabling practices — related to TD management into four quadrants. We design them inspired by the SWOT analysis [Shahir et al. 2008]. But unlike SWOT, the scope of the IDEA diagrams is not organizational planning but is to support software teams in increasing their ability to manage debt items. The diagrams can be defined for any TD management activity, and their practices and PAR can be specialized considering the types of debt (such as code, design, and requirements) and project context variables, such as the process model.

Figure 5 presents the diagram's structure and how the quadrants are related to each other. Each quadrant is depicted by a specific color and contains a set of practices or PAR. On the left side of the diagram, practices are concentrated in the actions and enabling practices quadrants. Actions (in the upper left quadrant) are practices or techniques that, when employed, will have a direct effect on TD management. Enabling practices (lower left), on the other hand, have an indirect effect on a team's ability to effectively manage TD by enabling a culture that promotes TD management or providing resources that are important for effective TD management. On the right side, the diagram presents the PARs

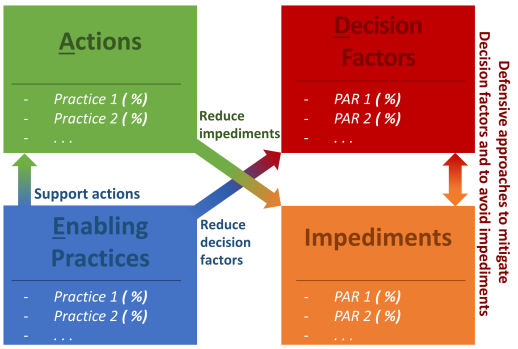

Figure 5. The IDEA diagram's structure

Figure 6. A summarized version of the IDEA diagram for design debt payment

in the decision factors and impediments quadrants. The decision facts (in the upper right quadrant) represent factors that led to decisions explicitly made by the team itself to incur TD or to not pay off TD. Impediments (lower right) are conditions or decision originating from an external agent (i.e., a customer or organization) who are outside the control of the project team, but that make it difficult or impossible to manage TD effectively. In all quadrants, the practices and PARs are ordered by a criterion that can be defined by software teams. For example, a sorting criterion could be how frequently practices and PARs have been used in the project in the past.

We used data from the InsighTD project to define IDEA diagrams for TD prevention, monitoring, and payment. Also, we specialized them for design and documentation debt. Figure 6 shows the IDEA diagram for design debt payment with the five most cited elements per quadrant. All complete diagrams are available in [Freire 2023]. The percentages with practices and PARs inform how frequently they were used in the InsighTD participants' software projects.

IDEA diagrams can support the definition of TD management strategies by analyzing one or two quadrants simultaneously. When looking at isolated quadrants, software teams can identify the actions used to manage the debt (actions quadrants) and the practices that support these actions (enabling practices quadrants) shown on the left of the diagram. Further, software teams can identify the issues that hamper TD management from decisions made by the team (decision factors quadrant) or by an external factor (impediments quadrant).

Analyzing the relationships between quadrants can support software teams in boosting their TD management initiatives. Consider Figure 6 as an example:

- **Actions and Enabling practices quadrants** can provide teams with a way to increase their TD management ability by suggesting other practices that could be implemented. For example, a software team that uses *code refactoring* and *design refactoring* actions to pay off design debt items can use *investing effort on TD payment activities* and *negotiating deadline extension* enabling practices to support these actions.

- **Decision factors and Impediments quadrants** can support teams in understanding why they are not managing TD. For example, a software team can identify that *focusing on short term goals* and *lack of testing* decision factors and *customer decision* impediment are the reasons for not paying off design debt items.
- **Enabling practices and Decision factors quadrants** can reduce weak areas related to TD management. For instance, if a team realizes that *lack of adoption of lessons learned* decision factor is the reason for design debt non-payment, the team can apply *improving software development process* and *improving the team collaboration* to change the team's mindset.
- **Actions and Impediments quadrants** can help teams to reduce the impediments for TD management. For example, if a team identifies that *complexity of the project* impediment hampers the payment of design debt, the team can apply *code refactoring*, *design refactoring*, and *adjusting the code to follow good programming practices* actions for reducing external factors in TD payment decisions.

4.2.1. Assessing the Diagrams

This section offers the complementary studies we performed to assess the IDEA diagrams in academic and industrial settings.

First Study: Assessing the ease of use, usefulness, and potential future use of the IDEA diagrams. The goal of this study is to **analyze** the IDEA diagrams **with the purpose of** characterizing them **with respect to** ease of use, usefulness, and potential future use **from the point of view of** undergraduate students enrolled in a software engineering course **in the context of** software development projects. As our intention is to investigate the perception on the use of a new technology (IDEA diagrams), we conducted the evaluation by applying the technology acceptance model (TAM) [Davis 1989]. It captures the participants' opinions on three constructs (perceived usefulness, ease to use, and self-predicted future use), measured by a set of questions.

The study consisted of analyzing the ease of use, usefulness, and potential future use of the IDEA diagrams through the simulation of TD management activities, whose objective was to identify, from a list of TD items, the prevention, monitoring, and payment practices and PAR that could be applied for the project.

In total, 72 students participated in the study and individually completed the evaluation form, containing a set of questions associated with the three constructs (perceived usefulness, ease to use, and self-predicted future use) considered in the TAM. To answer the questions in the form, the participants indicated the option that best represented their point of view on the IDEA diagrams, according to the following 5-point scale: (1) Strongly Agree - SA; (2) Partially Agree - PA; (3) Neutral - N; (4) Partially Disagree - PD; and (5) Strongly Disagree - SD. At the end of the form, the participants described the positive and negative aspects of the diagrams and suggestions for improvements and indicated whether the diagrams helped them to identify practices and PARs that they would not have identified without using them.

Regarding the **perceived usefulness** construct, most of the participants agreed with the affirmations for IDEA diagrams for TD prevention (more than 86% of the participants), TD monitoring (more than 81%), and TD payment (more than 89%). Thus,

comparing the task execution with and without IDEA diagrams, the participants had high productivity, increased performance, and efficacy. Moreover, 90% of the participants agreed with the following statements: "using the diagrams, I would increase my productivity in identifying practices and PARs" (SA: 65%, PA: 25%, and N: 10%) and "I believe the proposed diagrams would be useful to support technical debt management" (SA: 72%, PA: 18%, and N: 10%).

About the **ease-of-use** construct, at least 80% of the participants agreed with the statements associated with the benefits: easy to learn, clear and understandable, easy to use for particular tasks, easy to become skillful, easy to remember, and easy to use.

Concerning the **self-predicted future use** construct, 92% of the participants agreed with "Assuming the proposed diagrams would be available to manage technical debt, I would use them in the future" (SA: 63% and PA: 29%) and 63% of the participants agreed with "I would prefer to use the proposed diagrams to identify practices and PARs associated with TD prevention, monitoring and payment activities than in the usual way (without the diagrams)." Only 15% of the participants disagreed with this statement (SA: 45%, PA: 18%, N: 22%, PD: 7%, and PD: 8%).

Second Study - Perception of Software Practitioners. The goal of this study is to **analyze** the IDEA diagrams **with the purpose of** characterizing them **with respect to** their support to TD management activities **from the point of view of** software practitioners with experience in their roles **in the context of** software development projects.

We conducted semi-structured eleven individual interviews composed of three steps. In the first step (**opening**), we presented the consent form and the concept of TD. Then, the participant answered questions on TD management, such as the level of experience with TD management and the strategies and tools used to manage the debt. In the second step (**perception about the IDEA diagrams**), we presented the IDEA diagrams and provided some examples of using the diagrams for supporting TD management. Then, we asked participants whether the diagrams (i) are easy to use and follow, (ii) could influence their decision about how to manage the debt, and (iii) could be used in their daily activities. In the last step (**closing**), we asked participants if they had anything more to say about the diagrams and asked them to fill in a characterization form. In total, we interviewed 11 practitioners from our contacts in the software industry.

The results showed that most of the participants (nine participants) affirmed that the IDEA diagrams are easy to read and follow to support decisions on TD management because the diagrams: (i) facilitate TD decision making, (ii) are succinct and clear, (iii) can be understood by all stakeholders, (iv) present in a summarized way both internal and external issues, (v) can be used in reviewing and planning meetings, and (vi) facilitate TD items identification. Lastly, three of these participants warned that the diagrams are easy to use but are not self-explanatory.

Regarding the influence decision about how to manage the debt, only one participant indicated that the diagram would not influence his/her decisions. The other participants reported that the diagrams could influence their decisions, highlighting that the diagrams (i) facilitate the communication between stakeholders, (ii) support the decision making on TD items, (iii) support to identify problems, (iv) have a customizable catalogue of practices used in the software industry, and (v) allow an effective risk management. The

participants have different opinions on the percentages. Almost all participants (nine) indicated that the percentages would be useful for choosing a practice or a PAR, but other two participants mentioned that percentages can be difficult to calculate.

About the diagram can be used in daily activities, all participants indicated that they could use the IDEA diagrams to support TD management activities. They explained that the diagrams (i) enable continuous improvement of TD management actions, (ii) assist in tracking TD items, (iii) indicate possible problems and solutions to resolve them, and (iv) assist team communication. Most of the participants (six) indicated that the diagrams could be adapted to their current context because they would assist in negotiating project constraints and highlight the problems. The participants also indicated the following necessary adjustments in the diagrams: (i) remove practices that do not fit the developer's scope, (ii) include arrows between quadrants to indicate how the analysis should be done, and (iii) make it automated by suggesting relationships between quadrants.

Discussion. In summary, the TAM study and the interviews provided positive evidence that the IDEA diagrams can be useful for supporting TD management activities. Results also provide initial evidence that the IDEA diagrams can be used by practitioners with or without experience in managing TD items. For software teams who want to start managing TD, the ranked lists of practices and PARs organized in each of the IDEA diagrams can provide 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 IDEA diagrams.

5. Concluding Remarks

This section discusses the main scientific contributions, academic/professional impact, and future work.

5.1. Scientific Contributions

Regarding the main contributions related to the Ph.D. dissertation's specific goals, we initially reviewed the current state of research on TD management, allowing for identifying key research findings that addressed TD prevention, monitoring, and payment, meeting SG1. Specific contributions are described in Chapter 2 of the dissertation [Freire 2023].

Other contribution is the organization of an open and generalizable set of empirical evidence on TD prevention, monitoring, and payment collected from the InsighTD project and the analysis and synthesis of results from this set of empirical evidence, reaching SG2 and answering RQ1-RQ3. It generated (i) a comprehensive list of TD prevention, monitoring, and payment practices used by software practitioners to manage TD in their software projects, (ii) a comprehensive list of PARs that justify the non-application of prevention, monitoring, and payment practices in software development projects, and (iii) an analysis of prevention, monitoring, and payment practices and PARs that reveal their types, nature, categories, and relationships with types of debt. These results are detailed in Chapters 4-6 of the dissertation [Freire 2023].

The body of knowledge generated in SG2 was organized into three artifacts: (i) an updated version of the TD conceptual model, including the concepts from the state of

Table 1. Publications made during the doctorate

		Conference or Journal	Authorship	Qualis*	Reference**
Primary	1	ACM Transactions on Software Engineering and Methodology	First author	A3	[FREIRE et al., 2024a]
	2	The Journal of Systems and Software	First author	A2	[FREIRE et al., 2023a]
	3	IEEE Software	First author	A4	[FREIRE et al., 2021c]
	4	Journal of Software Engineering and Research Development	First author	B1	[FREIRE et al., 2024b]
	5	SANER	First author	A1	[FREIRE et al., 2021a]
	6	ACM SAC	First author	A2	[FREIRE et al., 2020c]
	7	EASE	First author	A3	[FREIRE et al., 2020b]
	8	PROFES	First author	A4	[FREIRE et al., 2023c]
Secondary	9	Information and Software Technology	Co-author	A2	[PÉREZ et al., 2021]
	10	The Journal of Systems and Software	Co-author	A2	[RAMAČ et al., 2022a]
	11	IEEE Software	Co-author	A4	[RIOS et al., 2021]
	12	IEEE Software	Co-author	A4	[MANDIĆ et al., 2021]
	13	Journal of Software Engineering and Research Development	Co-author	B1	[BERENGUER et al., 2023]
	14	AMCIS	Co-author	A2	[ROCHA et al., 2021]
	15	AMCIS	Co-author	A2	[BERENGUER et al., 2021a]
	16	CHASE	First author	A3	[FREIRE et al., 2021b]
	17	SBES	Co-author	A3	[SOUZA et al., 2020]
	18	TechDebt	Co-author	B1	[PÉREZ et al., 2020]
	19	SBQS	Co-author	B1	[BERENGUER et al., 2021b]
	20	SBQS	Co-author	B1	[SOARES et al., 2022]
	21	SAST	Co-author	B1	[ROCHA et al., 2022]
	22	WER	Co-author	B4	[BARBOSA et al., 2022]
	23	IS	Co-author	-	[RAMAČ et al., 2022b]
Related	24	The Journal of Systems and Software (no prelo)	Co-author	A2	[BARBOSA et al., 2024]
	25	SBES	First author	A3	[FREIRE et al., 2019]
	26	SBES	Co-author	A3	[GAMA et al., 2020]
	27	Euromicro SEAA	First author	A4	[FREIRE et al., 2020a]
	28	REFSQ	First author	A4	[FREIRE et al., 2023b]
	29	TechDebt	Co-author	B1	[GOMES et al., 2022]
	30	SBQS	Co-author	B1	[SANTOS et al., 2022]

* Qualis values from PPGCC/PUCRS. Accessed on 08/09/2024.

* The complete references are available here.

practice, (ii) a set of conceptual maps that organize the body of knowledge on practices and PARs associated with TD prevention, monitoring, and payment, and (iii) a set of IDEA diagrams that organize the body of knowledge on practices and PARs associated with TD prevention, monitoring, and payment. Theses artifacts support meeting SG3 and answering RQ4. Chapters 7 and 8 give more details.

Lastly, we performed three empirical studies (SG4) to assess the artifacts. As contributions, we can cite: (i) the empirical evidence about the accuracy and completeness of the updated version of TD conceptual model in relation to the software practitioners' perception about its representation of TD payment concepts, (ii) the empirical evidence about the accuracy and completeness of the TD payment map in software practitioners' point of view, (iii) the empirical evidence about the IDEA diagrams' support to TD management activities derived from two complementary empirical studies, one performed in an academic setting and another in the software industry.

5.2. Academic/Professional Impact

Table 1 presents all publications carried out during the doctorate, indicating their category, location, authorship and reference. The categories are as follows: primary, a publication that is in the scope of the Ph.D.; secondary, a publication that is not in the scope of the Ph.D. but is in the scope of the InsighTD project; and related, another kind of publication.

Looking at Table 1, one can notice that we have published eight papers that are direct results of this Ph.D. dissertation, being one in the ACM Transactions on Software Engineering and Methodology, one in the Journal of Systems and Software, one in the IEEE Software, one just accepted in the Journal of Software Engineering and Research Development, and four in conferences in the area.

Of the other twenty-one papers published during the period, fifteen are studies derived from this Ph.D. dissertation. Of these, three papers were coordinated by the Serbian InsighTD replication team (lines 10, 12, and 23 at Table 1) and two by the Colombian InsighTD replication team (lines 9 and 18). The other sixteen papers are related to model smells, causes and effects of TD, or software requirements (lines 16, 25, 27, and 28) or resulted from Master thesis, Ph.D. dissertations, and scientific works related to software engineering TD (lines 11, 13-15, 17, 19-22, 26, 29, and 30). Lastly, we have a paper submitted in the Journal of Systems and Software in the 2nd round of review (line 24).

During this period, we received four awards: (i) Distinguished Paper of the Brazilian Symposium on Software Quality - SBQS 2021 (line 19 at Table 1); (ii) Best Paper of the Workshop on Requirements Engineering - WER 2022 (line 22); (iii) Distinguished Paper of the Brazilian Symposium on Software Quality - SBQS 2022 (line 30), and (iv) Best Presentation in the Workshop de Estudantes de Pós-Graduação em Ciência da Computação do PGCOMP-UFBA (WEPGCOMP 2022).

The author of this Ph.D. dissertation has acted as an informal co-adviser of four former master's students and served a reviewer for peer-review conferences (Industry Track of X Brazilian Conference on Software: Theory and Practice 2019; Encontro Nacional de Computação dos Institutos Federais - ENCompIF 2021, 2022) and journals (Software Quality Journal; Journal on Interactive Systems; International Journal of Agile Systems and Management; and *Revista Eletrônica de Iniciação Científica em Computação*).

5.3. Future Work

Considering this Ph.D. dissertation as a starting point, we suggest the following prospective future work to: (i) conduct empirical studies to digest more the practices and PARs; (ii) conduct systematic reviews to identify what practices were previously investigated; (iii) investigate the correlations between practices and between PARs; (iv) study the cooccurrence between causes of TD and TD prevention and monitoring, and between effects of TD and TD payment; (v) plan and perform empirical studies to assess the complete conceptual model for TD and prevention and monitoring conceptual maps; (vi) plan and perform case studies to assess IDEA diagrams in industrial settings; (vii) automatize the IDEA diagrams; and (viii) define a strategy for TD management by investigating how conceptual maps and IDEA diagrams can comprise a strategy to support TD management, considering how software practitioners can identify the practices and PARs used/presented in their projects.

Acknowledgments

This study was financed in part by CAPES - Finance Code 001 and CNPq, Brazil.

References

Apa, C., Solari, M., Vallespir, D., and Travassos, G. H. (2020). A taste of the software industry perception of technical debt and its management in uruguay: A survey in software industry. In *Proc. of the 14th ESEM*, New York, NY, USA. ACM.

Aragão, B. S., Andrade, R. M. C., Santos, I. S., Castro, R. N. S., Lelli, V., and Darin, T. G. R. (2022). Testdcat 3.0: Catalog of test debt subtypes and management activities. *Software Quality Journal*, 30(1):181–225.

Basili, V. R. (1993). The experimental paradigm in software engineering. In Rombach, H. D., Basili, V. R., and Selby, R. W., editors, *Exp. Software Engineering Issues: Critical Assessment and Future Directions*, pages 1–12, Berlin, Heidelberg. Springer.

Basili, V. R., Selby, R. W., and Hutchens, D. H. (1986). Experimentation in software engineering. *IEEE Trans. Softw. Eng.*, 12(7):733–743.

Brown, N., Cai, Y., Guo, Y., Kazman, R., Kim, M., Kruchten, P., Lim, E., MacCormack, A., Nord, R., Ozkaya, I., Sangwan, R., Seaman, C., Sullivan, K., and Zazworka, N. (2010). Managing technical debt in software-reliant systems. In *Proc. of FoSER '10*, page 47–52, New York, NY, USA. ACM.

Cartaxo, B., Pinto, G., Vieira, E., and Soares, S. (2016). Evidence briefings: Towards a medium to transfer knowledge from systematic reviews to practitioners. In *Proc. of the 10th ESEM*, New York, NY, USA. ACM.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3):319–340.

Ernst, N. A., Bellomo, S., Ozkaya, I., Nord, R. L., and Gorton, I. (2015). Measure it? manage it? ignore it? software practitioners and technical debt. In *Proceedings of the 10th ESEC/FSE*, page 50–60, New York, NY, USA. ACM.

Freire, E. S. S. (2023). *Organizing the state of practice on technical debt prevention, monitoring, and payment in software projects*. PhD thesis, UFBA.

Guo, Y., Spinola, R. O., and Seaman, C. (2016). Exploring the costs of technical debt management — a case study. *Empirical Softw. Engg.*, 21(1):159–182.

Li, Z., Avgeriou, P., and Liang, P. (2015). A systematic mapping study on technical debt and its management. *J. Syst. Softw.*, 101(C):193–220.

Martini, A., Besker, T., and Bosch, J. (2018). Technical debt tracking: Current state of practice: A survey and multiple case study in 15 large organizations. *Science of Computer Programming*, 163:42–61.

Rios, N., de Mendonça, M. G., and Spinola, R. O. (2018). A tertiary study on technical debt: Types, management strategies, research trends, and base information for practitioners. *Information and Software Technology*, 102:117–145.

Rios, N., Spínola, R. O., Mendonça, M., and Seaman, C. (2020). 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.*, 25(5):3216–3287.

Rios Alves, N. S. (2020). *Organização de um conjunto de descobertas experimentais sobre causas e efeitos da dívida técnica através de uma família de surveys globalmente distribuída*. PhD thesis, UFBA.

Shahir, H. Y., Daneshpajouh, S., and Ramsin, R. (2008). Improvement strategies for agile processes: A swot analysis approach. In *2008 Sixth International Conference on Software Engineering Research, Management and Applications*, pages 221–228.

Zazworka, N., Spínola, R. O., Vetrò, A., Shull, F., and Seaman, C. (2013). A case study on effectively identifying technical debt. In *Proc.of the 17th EASE*, page 42–47, New York, NY, USA. ACM.