Journal of Software Engineering Research and Development, 2024, 12:11, doi: 10.5753/jserd.2024.4011

Hearing the Voice of Software Practitioners on Technical Debt Monitoring: Understanding Monitoring Practices and the Practices' Avoidance Reasons

Sávio Freire [Federal Institute of Ceará and Federal University of Bahia | savio.freire@ifce.edu.br]
Nicolli Rios [Federal University of Rio de Janeiro | nicolli@cos.ufrj.br]
Boris Pérez [University of Los Andes and Francisco de Paula Santander University | br.perez41@uniandes.edu.co]
Camilo Castellanos [University of Los Andes | cc.castellanos87@uniandes.edu.co]
Darío Correal [University of Los Andes | dcorreal@uniandes.edu.co]
Robert Ramač [University of Novi Sad | ramac.robert@uns.ac.rs]
Vladimir Mandić [University of Novi Sad | vladman@uns.ac.rs]
Nebojša Taušan [University of Novi Sad | nebojsa.tausan@ef.uns.ac.rs]
Gustavo López [University of Costa Rica | gustavo.lopezherrera@ucr.ac.cr]
Alexia Pacheco [University of Costa Rica | alexia.pacheco@ucr.ac.cr]
Manoel Mendonça [Federal University of Bahia | manoel.mendonca@ufba.br]
Davide Falessi [University of Rome Tor Vergata | falessi@ing.uniroma2.it]
Clemente Izurieta [Montana State University and Idaho National Laboratories | clemente.izurieta@montana.edu]
Carolyn Seaman [University of Maryland Baltimore County | cseaman@umbc.edu]
Rodrigo Spínola [Virginia Commonwealth University | spinolaro@vcu.edu]

Abstract

Context. Technical debt (TD) monitoring allows software professionals to track the evolution of debt incurred in their projects. The technical literature has listed several practices used in the software industry to monitor indebtedness. However, there is limited evidence on the use and on the reasons to avoid using these practices. **Aims**. This work aims to investigate, from the point of view of software practitioners, the practices used for monitoring TD items, and the practice avoidance reasons (PARs) curbing the monitoring of TD items. **Method**. We analyze quantitatively and qualitatively a set of 653 answers collected with a family of industrial surveys distributed in six countries. **Results**. Practitioners are prone to monitor TD items, revealing 46 practices for monitoring the debt and 35 PARs for explaining TD non-monitoring. Both practices and PARs are strongly associated with planning and management issues. The study also shows the relationship found among practices, PARs and types of debt and presents a conceptual map that relates practices and PARs with their categories. **Conclusion**. The results of this study add to a practitioners' capability to monitor TD items by revealing the monitoring practices, PARs and their relationship with different TD types.

Keywords: *Technical debt, Technical debt monitoring, Technical debt management, Survey*

1 Introduction

Technical debt (TD) items are the result of shortcuts or even mistakes taken by software practitioners during the development of software projects (Avgeriou *et al*. 2016; Falessi and Kazman, 2021). These items can bring benefits, like increased development speed, or drawbacks, such as extra cost and financial loss (Izurieta *et al*. 2012; Seaman *et al*. 2012; Curtis *et al*. 2012; Guo *et al*. 2014), e.g., on average 25% of the development time is spent on TD related activities (Curtis *et al*. 2012).

TD management intends to balance those short-term benefits and eventual long-term drawbacks of debt presence, keeping them visible and under control (Li *et al*. 2015). There are several activities associated with managing TD items (Li *et al*. 2015; Rios *et al*. 2018b). Amongst them, TD monitoring is a central activity that allows software teams to track unresolved debt items, identifying changes in their cost and benefit during the project life cycle (Rios *et al*. 2018b). For example, practitioners can continuously compare the actual costs of a debt item with the current benefits of its presence by using a cost/benefit analysis (Ampatzoglou *et al*. 2015).

TD monitoring is a topic that has already been investigated for some time. A tertiary (Rios *et al*. 2018b) and three secondary studies (Ampatzoglou *et al*. 2015; Li *et al*. 2015; Behutiye *et al*. 2017) have presented strategies and tools for supporting the monitoring of debt items. For example, SonarQube [1] TD plugin and TD evaluation (SQALE[2]) are tools that have been developed to support the enactment of TD monitoring. Martini (2018) and Mendes *et al*. (2019) also proposed tools for supporting TD monitoring. Further, cost/benefit analysis, real options analysis, and portfolio management are strategies used for monitoring the debt (Ampatzoglou *et al*. 2015). The studies of Ernst *et al*. (2015), Yli-Huumo *et al*. (2016), Martini *et al*. (2018), and Apa *et al*. (2020) investigated the practices used for monitoring debt items in general, while Oliveira *et al*. (2015) and Aragão *et al*. (2021) considered practices for monitoring design and defect, and test debt items, respectively. However, those studies were performed considering the

[1] https://www.sonarqube.org/

[2] http://sqale.org/

context of TD management as a whole, i.e., none of them focused specifically on TD monitoring, and the results reported by them are grounded on small sample sizes or few organizations.

Despite current research efforts on the topic, the discussion around TD monitoring deserves a more comprehensive investigation. The main reason for this is that TD monitoring has the key role in preventing TD from spreading through the project which leads to increased development costs or, in some extreme cases, project termination (Rios *et al*. 2020b). This is because the software development process is a chain of interconnected activities and decisions, where one activity relies on the artefacts produced by others. If the artefact produced has TD, it is passed on to all activities that rely on it (Mandić *et al*. 2021). Thus, it is important to continuously evaluate whether the costs of TD overcome its benefits and to act accordingly. Furthermore, we know that a multitude of factors lead to situations favorable for injecting TD during the software development and maintenance process, e.g., factors like *time pressure*, *lack of knowledge*, etc. (Mandić *et al*. 2021; Ramač *et al*. 2022a). It is evident that factors can be associated with different aspects of an organization, as well as factors that are external to the organization. Therefore, to gain the insights of the TD monitoring practices and TD monitoring *avoidance reasons*, we also need to look at associated contexts of the practices. For example, context can be infrastructure, people, process, or external factors.

We also have investigated TD monitoring in our prior studies (Rios *et al*. 2021; Freire *et al*. 2021c) using data from the *InsighTD* project, which investigates the state of the practice of TD (causes, effects, and management of TD) by conducting a family of surveys applied in software industries in several countries (Rios *et al*. 2020b). In Rios *et al.* (2021), we investigated whether the process model adopted by software teams influences the management (prevention, monitoring, and repayment) of TD. We found that agile teams are more inclined to monitor TD items than those using other process models. In another work (Freire *et al.* 2021c), based on the responses from 274 practitioners, we proposed the IDEA diagrams for organizing practices and practice avoidance reasons (PARs) related to TD prevention, repayment, and monitoring in agile software projects. The IDEA diagrams organize issues (decision factors and impediments) and capabilities (actions and enabling practices) into four quadrants to support software teams concerning TD prevention, monitoring, and repayment. More details are available in Freire *et al.* (2021c). The diagrams contained only the ten most-cited TD monitoring practices and PARs reported by agile software practitioners. We also assessed the diagrams for supporting TD management (Freire *et al*. 2024a).

Although our prior studies (Rios *et al*. 2021; Freire *et al*. 2021c) provide preliminary results on TD monitoring, much remains to be investigated. We performed these studies on more restricted data sets than is being used in this study. By using a larger and broader data set, we build support to generalize our previous results, as we have answers provided by practitioners from different development cultures, i.e., from different roles, following different process models, and from different countries. Furthermore, we can increase the ecological validity (Andrade 2018), as well as the confidence of our results (Wohlin *et al*. 2012).

Further, having a larger and broader data set allows us to perform different analysis to support software teams in designing their own TD monitoring strategy. For example, the relation between the TD monitoring practices and PARs with types of debt (as we have included in our analysis) can reveal specific practices to monitor a type of debt affecting a software project or specific PARs to increase the software team's ability in monitoring this type of debt. Also, familiarity with TD monitoring practices can help practitioners when deciding which practice(s) should be used in their projects and contexts. Comprehension of PARs, on the other hand, supports practitioners in identifying issues that curb TD monitoring and seek to eliminate or mitigate them from their context. These PARs can avoid the application of a specific practice or indicate that a specific practice is missing in a project. By organizing this body of knowledge, we can provide valuable artifacts that practitioners and researchers can use as starting point to perform their activities. Having information from the software industry about practices and PARs can drive new research efforts on tools and strategies (combination of practices) for TD monitoring aligned with the practitioners' needs.

In this context, this work aims *to investigate, from the point of view of software practitioners, the practices used for monitoring TD items and the PARs curbing the employment of those practices*. To this end, we use the full *InsighTD* data set, which encompasses answers from 653 practitioners from Brazil, Colombia, Chile, Costa Rica, Serbia, and the United States. This work significantly extends our prior work on the topic by presenting:

(1) The updated and full list of identified practices ranked by the most cited. We also found that practitioners see the monitoring of debt items as an integrated part of a more comprehensive TD management strategy. The identified practices can vary from those used specifically to monitor debt items, to practices for enabling TD monitoring, and repaying, preventing, and identifying debt items.

(2) The updated and full list of identified PARs organized by the most cited. PARs can be a decision factor or an impediment, revealing that a team can deliberately decide to not monitor TD or face a situation that impedes the use of monitoring practices.

(3) The identification of the main groups of issues—*categories*—associated with TD monitoring. The identified categories help us to better contextualize related factors. Practices related to the project management and the methodology employed in projects are the most commonly cited, indicating the importance of managerial and methodological issues on TD monitoring. Similarly, we found indications of the influence of managerial issues in TD non-monitoring.

(4) The relationships between types of debt and practices, and types of debt and PARs, indicating which practices have been used to monitor specific types of debt or which PARs explain the TD non-monitoring when practitioners deal with specific TD types.

(5) Definition of a conceptual map organizing the practices and PARs per categories and types. The map can be used as a conceptual guide, supporting practitioners in choosing practices or in identifying PARs experienced in practice. Also, researchers can use the map to understand and guide new research efforts considering the state of the practice in TD monitoring.

Besides the introductory section, this paper is organized into the following six sections. Section 2 presents the background, addressing TD monitoring and related work, as well as the work on TD monitoring in the *InsighTD* project. Section 3 presents the research method. Section 4 presents the results. Section 5 discusses the study's main findings, presents the conceptual map, and compares the findings with those reported in related work. Section 6 discusses the threats to the study validity. Lastly, Section 7 presents our final remarks and the next steps of this work.

2 Background and Related Work

TD management is decisive for increasing the success of software projects (Seaman and Guo 2011). It is composed of the following activities: identification, measurement, prioritization, prevention, monitoring, payment, documentation, communication, visualization, time-to-market analysis, and scenario analysis (Li *et al*. 2015; Rios *et al*. 2018b). Among these activities, TD monitoring allows software teams to observe the changes in the cost and benefit of unresolved debt items during the project's evolution (Li *et al*. 2015; Rios *et al*. 2018b). TD monitoring has a set of practices which support software teams in tracking TD items in their projects. If well monitored, the TD cost is kept visible and under control, supporting the project to achieve its goals sooner or cheaper. On the other hand, TD items can cause financial and technical issues, compromising the project's future (Rios *et al*. 2020b). Teams need to be aware of TD items in their projects and apply practices and strategies for managing them (Kruchten *et al*. 2012).

2.1 TD monitoring practices

In the technical literature, several practices have been proposed for supporting TD monitoring (Ampatzoglou *et al*. 2015; Li *et al*. 2015; Behutiye *et al*. 2017; Rios *et al*. 2018b). Amongst them, *cost/benefit analysis*, *real options analysis*, and *portfolio management* are the most popular approaches for monitoring the debt (Ampatzoglou *et al*. 2015). By *cost/benefit analysis,* software teams can analyze their TD items evolution, comparing cost and benefit for deciding on their repayment (Ampatzoglou *et al*. 2015). Performing *real options analysis,* practitioners can quantify the long-term value associated with a TD item, track it during the project, and decide how to deal with it (Alzaghoul and Bahsoon 2014). Lastly, *portfolio management* allows that, in an iterative way, TD items are assessed and tracked to decide if one of them should be repaid (Ampatzoglou *et al*. 2015).

Ampatzoglou *et al*. (2015) performed a systematic literature review to identify the financial aspects of TD. Through answering the research question "RQ2.2 Which financial approaches have been applied for identifying, prioritizing, repaying, and monitoring technical debt?," the authors identified the following approaches for TD monitoring: *accounting*, *cost/benefit*, *real options*, *marketing*, and *portfolio management*. Ernst *et al*. (2015) conducted a two-part study composed of a survey with practitioners in three large companies and semi-structured follow-up interviews for investigating, among others, the use of tools, and techniques to manage TD. Through the research question "Are there practices and tools for managing TD?," the authors identified that practitioners have applied TD monitoring *as part of risk process or backlog grooming*. Although our work also investigates the practices used to monitoring the debt, we seek to identify the PARs used to explain the TD non-monitoring and the relations between types of debt with practices and with PARs.

In another related work, Li *et al*. (2015) run a systematic mapping study to understand the state of the art on TD concept and its management. Through the research question "What approaches are used in each TDM activity?," they reported the categories *threshold-based approach*, *TD propagation tracking*, *planned check*, *TD monitoring with quality attribute focus*, and *TD plot* as associated with TD monitoring activity. Our work also categorizes the practices, but it also reports the PARs and their respective categories. Our list of practices and PARs tend to be closer to the TD monitoring's state of the practice, as they were identified from the practitioners' point of view. Oliveira *et al*. (2015) performed an action research study in two companies using Scrum for evaluating the application of the TD management framework proposed by Seaman and Guo (2011). This framework is composed of three stages: (i) to identify the list of TD items, (ii) to measure the effort for eliminating the items, and (iii) to monitor the items for deciding the most appropriate time to deal with them. Concerning TD monitoring, the authors found that *defining a responsible person for monitoring each identified and measured TD item* were used by both companies, but lack of tools can make the integration between TD measurements and tracking charts difficult. Alternatively, our study is not limited to only consider the point of view of agile software development practitioners.

Yli-Huumo *et al*. (2016) ran a case study in a large company. Through the research question "What methods, models, practices, or tools do the studied development teams use for each TDM activity?," the authors found that TD monitoring was conducted rarely, and *used data collected from (management or TD measuring) tools*. Our study also investigates the practices and the PARs for TD non-monitoring. In other work in the area, Behutiye *et al*. (2017) conducted a systematic literature review to investigate the state of the art on causes of TD, its consequences, and management strategies in the context of agile software development. By answering the research question "What are the strategies proposed in the literature to manage TD in agile software development?," the authors found that *collective dashboards*, *visualization techniques*, *continuous integration tools*, *setting a commonly agreed definition of done*, *improving estimation techniques of sprints*, *planning in advance for TD*, and *implementing pair programming or test-driven development* are practices used for monitoring TD items. Our study is not limited to agile software development practitioners' perception.

Martini (2018) presented a management tool, called AnaConDebt, to support tracking and assessing TD items. The tool allows *the creation of TD items in a backlog* for monitoring these items. Among its functionalities, AnaConDebt tracks TD items in a dedicated repository, where they can be characterized with attributes (such as name, description, etc.). Differently, our study does not present an automated tool for monitoring the debt, but we present an artifact (See Section 5.2) to support the choosing of practices and PARs. Martini *et al.* (2018) conducted a multi-method research study including a survey and a multiple case study in three companies to understand the effort spent by software practitioners for managing TD. Through the research questions "What tools are used to track TD?," "How do software organizations introduce a TD tracking process?," and "What are the initial benefits and challenges when large organizations start tracking TD?," the authors found that surveyed practitioners have used tools for monitoring TD. These tools support the following monitoring practices: *using comments in the code or other artifacts*, *documenting issues in text or spreadsheets*, *using a system for bug fixing*, *reporting TD items in the backlog*, *statically analyzing the code for finding TD items or potential bugs, or security issues*, and *measuring test coverage*. Our work also investigates the practices for TD monitoring, but it also presents the relation between these practices with types of debt.

Rios *et al.* (2018b) performed a tertiary study to investigate the current state of the art on TD and its management. By answering the research question "What are the activities, strategies and tools that have been proposed to support the management of TD?," the authors identified a set of tools that support TD monitoring, such as Sonar TD plugin, DebtFlag, TD evaluation (SQALE), software maps tool, and code Christmas tree. Also, the following practices were recognized for monitoring TD items: *accounting*, *cost-benefit analysis*, *options*, *SQALE method*, *debt symptoms index*, *metrics for managing architectural TD*, *RE-KOMBINE model*, *measuring symptom severity on a smell thermometer*, *making of dependencies and code problems*, *supply chain management*, *formal approach to TD decision making*, *portfolio approach*, and *marketing*. Although our study also investigates the practices for TD monitoring, it reports the PARs used to explain the TD non-monitoring. These practices and PARs are closer to the state of the practice, as they were drawn from practitioners' perceptions. More recently, Mendes *et al.* (2019) developed and evaluated a tool, named VisminerTD, for supporting TD identification and monitoring activities. The tool presents the set of TD items distributed in *three panels based on the Kanban concept*. Our work does not propose an automated tool but presents an artifact to support the TD monitoring initiatives.

Apa *et al.* (2020) surveyed software practitioners from Uruguayan software startup organizations to learn how these practitioners perceive and manage TD items in their projects. As a result, *manual monitoring* was performed by the practitioners, but *tools* (Jira and Wiki) have also been used for monitoring TD items. Our work is not limited to investigating TD monitoring practices. In other recent work in this area, Aragão *et al.* (2021) defined a catalog for test debt items and their management, which encompasses the following TD monitoring practices: *monitor changes in the cost/benefit ratio of the identified debt*, *monitor triggers*, and *changes in the test process (if the team has an existing test process)*. Our study takes into consideration the different types of debt reported in the technical literature (Rios et al. 2018b).

By analyzing the technical literature, we found evidence of TD monitoring practices. Table 1 summarizes relevant information about the related work, reporting whether they

Table 1. Limitations of related work on TD monitoring-related practices.

Related work	TD monitoring		Type of debt	Representativeness		Research method
	Practice	Category		Sample size	Number of organizations	
Ampatzoglou *et al.* (2015)	Yes	No	General	-	-	Systematic review
Ernst *et al.* (2015)	Yes	No	General	536	3	Case study
Li *et al.* (2015)	No	Yes	General	-	-	Systematic review
Oliveira *et al.* (2015)	Yes	No	Design and defect	16	2	Action research
Yli-Huumo *et al.* (2016)	Yes	No	General	25	1	Case study
Behutiye *et al.* (2017)	Yes	No	General	-	-	Systematic review
Martini (2018)	Yes	No	General	-	-	Tool development
Martini *et al.* (2018)	Yes	No	General	226	3	Case study
Rios *et al.* (2018b)	Yes	No	General	-	-	Systematic review
Mendes *et al.* (2019)	Yes	No	General	28	-	Case study
Apa *et al.* (2020)	Yes	No	General	33	-	Survey
Aragão *et al.* (2021)	Yes	No	Test	5	1	Case study

found monitoring practices or categories of them, types of debt covered by them, the representativeness of each study (sample size and the number of organizations), and the research method used. We notice that most of the studies had quite a small sample size, except for studies performed by Ernst *et al.* (2015) and Martini *et al.* (2018). However, both studies were performed in only three organizations, reducing diversity in their set of participants. We also highlight that most of the related work approach the monitoring of TD only as a side variable of the TD management, not as the main goal of the study, reducing the findings on the topic. Overall, the current literature on TD monitoring practices is limited in several ways. Moreover, we did not find any study on PARs to explain the non-adoption of TD monitoring practices.

Despite the aforementioned limitations, we recognize the importance of related work and will use those results to perform a comparison with the results of *InsighTD* and observe how they complement each other. The result of the comparison is described in Subsection 5.3.

2.2 Results from the *InsighTD* project and investigations of the TD monitoring practices

In the past years, several results from the *InsighTD* project have been published in various forums. Table 2 summarizes *InsighTD* publications with respect to the research focus and viewpoint considered during the data analysis. The studies are organized considering their published year. Concerning the TD management focus, Freire *et al.* (2020b) and Freire *et al.* (2024b) analyzed *InsighTD* data sets to understand TD prevention in the software industry, revealing a list of preventive actions used by software practitioners for avoiding TD items and the impediments faced by practitioners that hamper the application of those actions. Pérez *et al.* (2021) investigated the repayment and prevention practices used by software architects by analyzing 72 answers collected by Brazilian, Chilean, Colombian, and North American *InsighTD* replication teams in their respective software industries. Freire *et al.* (2020a), Pérez *et al.* (2020), and Freire *et al.* (2023) investigated how practitioners have repaid TD in their projects. In Freire *et al.* (2021c), we investigated the practices and PARs for TD prevention, monitoring, and repayment considering the agile software practitioners' point of view. Lastly, Freire *et al.* (2021b) investigated the relationship between the effects of TD and repayment practices.

Recently, we approached TD monitoring by investigating whether the process model influences TD monitoring (Rios *et al.* 2021) and the identified practices and PARs related to TD monitoring in agile software projects (Freire *et al.* 2021c) (cells in gray of Table 2). This work differs from those in several aspects:

1. we do not limit our data analysis to the context of agile software projects, which significantly expands our data set.
2. we go deeper into the data analysis by not limiting our discussions to the Top 10 ranked TD monitoring practices and PARs.
3. we investigate the relation between TD monitoring practices/PARs and types of debt.
4. we shed light on the main concerns software teams should have on TD monitoring by grouping the identified practices and PARs into higher level categories.

Table 2. Summary of the *InsighTD* publications. Symbols for used data set: (◦) data from a single replication, (○) data from several replications, and (●) data from all replications. Cells in gray indicate studies on TD monitoring carried out with *InsighTD* data sets.

			TD research focus					
No.	**Reference**	**Conference (C)/ Journal (J)**	**Concept**	**Causes/Effects**	**Payment/payment avoidance**	**Prevention**	**Monitoring**	**Viewpoint considered during analysis**
1	Rios *et al.* (2018a)	C		◦				Comprehensive
2	Rios *et al.* (2019a)	C		◦				Comprehensive
3	Rios *et al.* (2019b)	C		◦				Agile sw. projects
4	Perez *et al.* (2019)	C		◦				Comprehensive
5	Pacheco *et al.* (2019)	C	◦					Comprehensive
6	Rios *et al.* (2020a)	C		○				Documentation debt
7	Rios *et al.* (2020b)	J	◦	◦				Comprehensive
8	Ramač *et al.* (2022a)	J	●	●				Comprehensive
9	Mandić *et al.* (2020)	C	◦					Comprehensive
10	Ramač *et al.* (2020)	C		◦				Comprehensive
11	Souza *et al.* (2020)	C		◦				Software testing
12	Freire *et al.* (2020a)	C						Comprehensive
13	Freire *et al.* (2020b)	C				○		Comprehensive
14	Pérez *et al.* (2020)	C						Comprehensive
15	Berenguer *et al.* (2021)	C	●					Comprehensive
16	Freire *et al.* (2021a)	C		●				Experience level
17	Freire *et al.* (2021b)	C		○				Comprehensive
18	Rocha *et al.* (2021)	C		○		○		Software testing
19	Mandić *et al.* (2021)	J		●				Experience level
20	Pérez *et al.* (2021)	J			○	○		Software architects
21	Freire *et al.* (2021c)	J			●	●	●	Agile sw. projects
22	Rios *et al.* (2021)	J			○	○	○	Process models

23	Ramač *et al.* (2022b)	C			◦			Comprehensive
24	Barbosa *et al.* (2022)	C						Requirements debt
25	Rocha *et al.* (2022)	C		○		○		Software testing
26	Berenguer *et al.* (2022)	J	●					Comprehensive
27	Freire *et al.* (2023)	J			●			Comprehensive
28	Freire *et al.* (2024a)	C			●	●	●	Comprehensive
29	Freire *et al.* (2024b)	C				●		Comprehensive

3 Research Method

This section presents the *InsighTD* project, our research questions, data collection, and data analysis procedures.

3.1 The *InsighTD* survey

The *InsighTD* project is a globally distributed *family of industrial surveys* on TD. The project aims to organize an open and generalizable set of empirical data on the problems of TD faced by software practitioners in their projects (Rios *et al*. 2020b). Its design supports replication of the survey in different countries. To date, the project has concluded data collection for the replications in Brazil, Chile, Colombia, Costa Rica, Serbia, and the United States.

The *InsighTD* planning comprises four stages: conception, validation, initiation, and international replication. In the conception stage, the research questions, the survey design, and strategies for collecting and analyzing data were defined. For the validation stage, a set of validations (internal, external, and pilot study) were carried out to check the survey questions for clarity and completeness. Four experienced researchers from the *InsighTD* project performed internal validation, and an experienced researcher in the empirical software engineering and TD fields conducted an external validation. This researcher is not part of the project. After applying adjustments in the survey required from internal and external validations, a pilot study was run with five practitioners to identify vague questions and incomplete answers in closed questions. In the initiation stage, the first execution of the survey was performed in Brazil, after which an empirical package was made available allowing the replication of the survey by other *InsighTD* teams. Replication of the survey by other teams represents the fourth stage of planning the international replication. Further details about the design of the survey and data collection protocols can be found in: Rios *et al*. (2018a), Rios *et al*. (2020b), and Ramač *et al*. (2022a).

3.2 Research questions

Our research goal is *to investigate, from the point of view of software practitioners, the practices used for monitoring TD items and the PARs curbing the employment of those practices*. Based on it, we define the research questions (RQs) presented in Table 3. The goal of **RQ1** is to identify the practices used by software practitioners to monitor TD, along with their categories. By identifying the categories, we seek to verify whether the practices are related to each other considering software project concerns. Furthermore, identifying categories also help us to contextualize related factors. **RQ1.1** seeks to investigate which of the practices have been used for monitoring each type of debt. By answering this question, we can support the choice of monitoring strategies. For example, if a team wants to monitor requirements debt items, our subset of practices for this type can be used as a starting point for this activity.

Table 3. Research questions.

ID	Research Question (RQ)
RQ1	What are the leading practices for monitoring TD items in software projects?
RQ1.1	What are the practices used for monitoring each type of debt?
RQ2	What are the leading practice avoidance reasons to explain the non-monitoring of TD items?
RQ2.1	What are the main practice avoidance reasons for non-monitoring each type of debt?

RQ2 aims at identifying the PARs for not applying the TD monitoring practices and their categories. Having information about the categories of PARs can support software teams in recognizing PARs related to each other existing in software projects. **RQ2.1** seeks to investigate if the identified PARs are specific or shared among different types of debt. This investigation can reveal leading PARs that explain TD non-monitoring per type of debt, improving the practitioners' ability for monitoring TD items from a specific type.

3.3 Data collection

To answer the RQs, our study uses data collected in the context of the *InsighTD* Project. Although the questionnaire[1] comprises 28 questions (Rios *et al*. 2020b), we only use the subset of questions related to the participants' characterization and TD monitoring. Table 4 presents those questions, indicating the question identification number (No), description, and type (closed or open-ended question).

The survey's participants characterized themselves and their organization and project in questions Q1 through Q8 concerning the company's size they work (options: 1–10 employees, 11-50 employees, 51–250 employees, 251–500 employees, 501–1000 employees, 1001–2000 employees, and more than 2000 employees), the country they work (options: all countries in the world), the system's size (options: less than 10KLOC, 10–100KLOC, 100KLOC–1MLOC, 1–10MLOC, and 10+ MLOC), the system's age (options: less than 1 year, 1-2 years, 2-5 years, 5-10 years, and more than 10 years), the team's size (options: less than 5 people, 5–9 people, 10–20 people, 21–30 people, and more than 30 people), the role they play (options: business analyst, DBA/data analyst, developer, process analyst, project

[1] The questionnaire is available at https://goo.gl/zRwSGa.

Table 4. Subset of the *InsighTD* survey's questions related to TD monitoring (Adapted from Rios *et al.* 2020b).

No.	Question (Q) Description	Type
Q1	What is the size of your company?	Closed
Q2	In which country you are currently working?	Closed
Q3	What is the size of the system being developed in that project? (LOC)	Closed
Q4	What is the total number of people of this project?	Closed
Q5	What is the age of this system up to now or to when your involvement ended?	Closed
Q6	To which project role are you assigned in this project?	Closed
Q7	How do you rate your experience in this role?	Closed
Q8	Which of the following most closely describes the development process model you follow on this project?	Closed
Q10	In your words, how would you define TD?	Open
Q13	Give an example of TD that had a significant impact on the project that you have chosen to tell us about:	Open
Q15	About this example, how representative it is?	Closed
Q24	Once identified, was the debt item monitored?	Closed
Q25	If yes, how? If not, why?	Open

leader/project manager, requirements analyst, software architect, test manager/tester, and Other:), the experience in that role (options: novice - minimal or "textbook" knowledge without connecting it to practice, beginner - working knowledge of key aspects of practice, competent - good working and background knowledge of area of practice, proficient - depth of understanding of discipline and area of practice, and expert - authoritative knowledge of discipline and deep tacit understanding across area of practice), and the development process used in the project (options: agile - a lightweight process that promotes iterative development, close collaboration between the development team and business side, constant communication, and tightly-knit teams, hybrid - is the combination of agile methods with other non-agile techniques. For example, a detailed requirements effort, followed by sprints of incremental delivery, and traditional - conventional document-driven software development methods that can be characterized as extensive planning, standardization of development stages, formalized communication, significant documentation and design up front).

In Q10, the participants provided their definition of TD. Further, the participants described in Q13 a TD item that occurred in their project and indicated in Q15 how frequently this item occurred. The participants' point of view on TD monitoring was captured in Q24 and Q25. In the former, the participants revealed whether the TD item described by them in Q13 was monitored or not. In the latter, they explained how this item was monitored (**RQ1**) or why it was not monitored (**RQ2**). As the TD item described in Q13 can be associated with a type of debt, and the example provided in this question is the context for answering Q24 and Q25, we can identify the practices (**RQ1.1**) and the PARs (**RQ2.1**) related to each type of debt.

For collecting the responses in the Brazilian, Chilean, Colombian, Costa Rican, North American, and Serbian software industries, we used LinkedIn, industry-affiliated member groups, mailing lists, and industry partners as invitation channels. The data-gathering stage was done in 2018 in Brazil and the United States, and 2019–2020 in Chile, Colombia, Costa Rica, and Serbia.

We use two criteria for validating the collected answers. First, we check if the definition of TD (Q10) and the provided example of TD item (Q13) given by the participants are in conformance with the TD definition[1] used in the *InsighTD* project (Rios *et al.* 2020b). Second, we verify whether the answers given to Q25 are really related to TD monitoring, meaning that the answers reflect monitoring practices or PARs for TD non-monitoring. The collected answers that did not fit these criteria were not considered in our analysis.

3.4 Data analysis

We use different data analysis procedures as we have closed and open-ended questions in the survey instrument. We use descriptive statistics and calculate the share of participants choosing each option available in each closed question. These procedures are used in Q1 through Q8 (characterization questions), Q15 (the frequency that the TD item occurs in the project), and Q24 (whether the TD item was monitored or not).

For the open-ended questions, we apply qualitative data analysis techniques (Seaman 1999; Strauss and Corbin 1998) because we did not provide a predetermined list of practices and PARs for participants. In answers given to Q25, we apply manual-open coding resulting in a set of codes. These codes can represent practices used for monitoring TD items or PARs according to answers given to Q24 (yes/no question). When Q24 receives a positive answer, the code is a TD monitoring-related practice, supporting answering **RQ1**; otherwise, a practice avoidance reason (PAR), supporting the response to **RQ2**. For example, *TD item backlog*, *tracking TD items*, and *TD as a task* are

[1] The TD definition used in InsighTD project is adapted from McConnell (2007): "Technical debt contextualizes the problem of outstanding software development tasks (for example, tests planned but not executed, pending code refactoring, pending documentation update, use of bad design practices, code that does not exhibit good coding practices) as a kind of debt that brings a short-term benefit to the project (normally in terms of higher productivity or shorter release time of software versions), that may have to be paid later in the development process with interest (for example, a poorly designed class tends to be more difficult and costly to maintain than if it had been implemented good object-oriented practices)."

TD monitoring-related practices, while *focusing on short term goals*, *lack of interest*, and *effort* are PARs. After identifying the codes, we iteratively revised and unified them until resulting in a list of codes and their respective number of occurrences. Code unification required some effort. For example, participants cited the following explanation to TD non-monitoring: "*due to tight deadlines*," "*lack of time*," and "*the project timeline didn't allow it.*" The initially extracted PARs were *tight deadlines*, *lack of time*, and *insufficient timeline*, respectively. Then, as these PARs had different nomenclature but shared an ordinary meaning, we unified them as *lack of time*.

Table 5. Categories adapted from Rios *et al*. (2020b)

Name	Definition
Development issues	Refers to TD elements issues that occur during project development.
External factors	Encompasses TD elements that are external to the development team and organization.
Infrastructure	Groups TD elements related to tools, technologies, and development environments.
Internal quality issues	Encompasses TD items related to internal quality issues.
Lack of knowledge	Refers to TD elements related to the team's lack of knowledge to develop the project.
Methodology	Refers to TD elements related to processes and methodologies used in the development of the project.
Organizational	Groups TD elements associated with organizational level.
People	Encompasses TD elements directly related to members of software development teams.
Planning and management	Groups TD elements related to project planning and management.

In each *InsighTD* replication, at least three researchers conduct the coding process, assuming one of the following roles: (i) **code identifier**—responsible for extracting the existing codes in the answers, (ii) **code reviewer**—responsible for reviewing all extracted codes, and (iii) **referee**—responsible for resolving disagreements in codes identified by the code identifier and code reviewer. In total, we had six code identifiers, six code reviewers, and six referees. Further, the last author reviewed the codes collected in all replications, pursuing consistency among all analyses.

We calculated the agreement level among the coder identifiers and reviewers using Cohen's Kappa coefficient (McHugh, 2012). It results in a number κ between [−1, +1], indicating the highest disagreement and agreement, respectively. We obtained a κ = + 0.802, indicating a high agreement rate and reliability for the coding process performed for all InsighTD replication teams.

After analyzing the TD monitoring-related practices, we continued to apply open coding to further understand the dimensions related to the concept of TD monitoring, resulting in the following groupings:

- **Types of TD monitoring practices.** The data showed that practitioners have applied distinct types of TD management practices to support their TD monitoring initiatives. For example, while the practice tracking the cost allows TD monitoring, the practice team restructuring improves the team's ability to monitor TD and can come from other TDM activities, like prevention, identification, and repayment. Following the open coding phase, the first author identified the types of practices, which were reviewed by the last author. The divergences were resolved in a consensus meeting.
- **Categories of TD monitoring practices.** We realized that many of the practices are related to each other, thus we organized them into categories which are, in their turn, related to software development concerns. For example, we used the category planning and management to group the practices TD item backlog and tracking TD items. To name the categories, we used the categories proposed by Rios et al. (2020b) which are presented in Table 5. The first author grouped the practices into categories, and then, the last author reviewed the grouping. The divergences were resolved in a consensus meeting.

Lastly, we identified the type of debt associated with the TD item described by participants in Q13. For this, in each *InsighTD* replication, two researchers separately looked at the participants' answers (Q13) and mapped them to the definitions of TD types reported in Rios *et al.* (2018b) or to the list of TD indicators given in Alves *et al.* (2016). For example, a participant gave the following answer in Q13: "*inaccurate, insufficient, and inadequate definition of stories and tasks*." As it describes issues in the requirements (underlined part), it represents a scenario of requirements debt. Lastly, a third researcher settled eventual divergences. For answering **RQ1.1** and **RQ2.1**, we associated the type of debt identified in Q13 with the practices or PARs coded from Q25. This association is possible because participants answered the questions on TD monitoring (Q24 and Q25) taking the provided example in Q13 as context.

Figure 1 summarizes how our analysis process evolved to reveal the groupings described above. It shows that for TD monitoring practices and PARs, we used open coding to identify practices and PARs along with their types and categories by grouping them into different properties and dimensions (Strauss and Corbin 1998). Lastly, we investigated the relationships between practices and types of debt, and PARs and types of debt.

4 Results

This section presents the results, describing the demographics data and answers to research questions. The

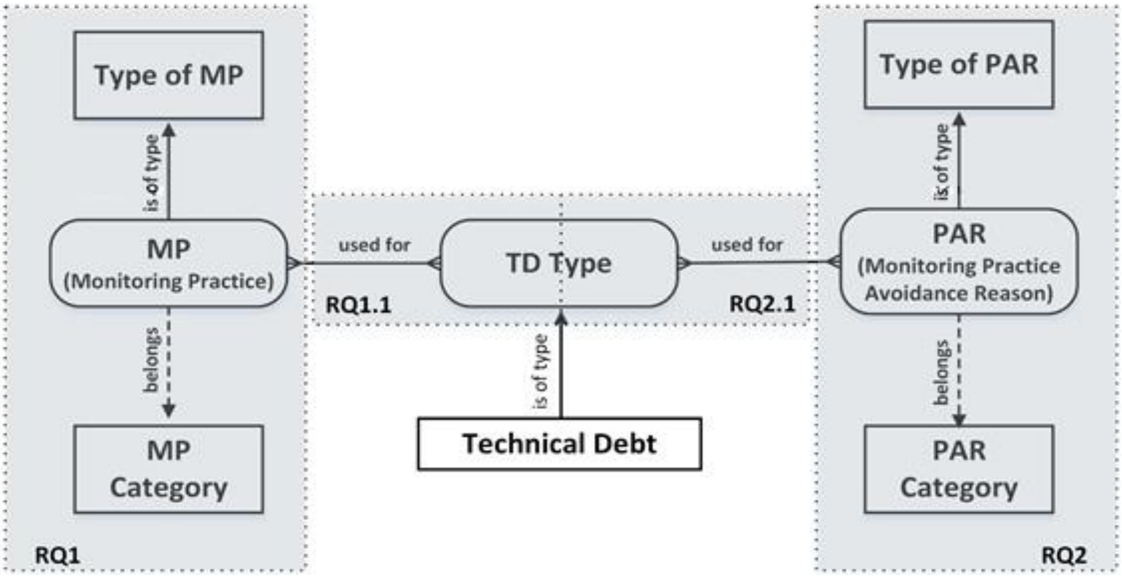

Figure 1. Relationship between the research questions and their findings.

answers to RQs are structured from two parts, and these parts present: (a) identified practices and PARs along with their respective types and categories and (b) the relation of TD types with the practices, PARs and their categories.

4.1 Demographics

In total, we collected 745 answers from the survey, but 12% of them were discarded as they did not fit in the validation criteria previously explained in Section 3.3. Thus, we have 653 valid answers[1] from the survey's participants (Brazil: 107 answers, Chile: 89, Colombia: 134, Costa Rica: 145, Serbia: 79, and the United States: 99). Figure 2 presents a dashboard summarizing the characterization of the participants.

The participants commonly work in medium-sized companies (51 to 1000 employees; 39%), but we found participants that work in small (up to 50 employees; 32%) and large (more than 1000 employees; 29%) organizations too. Further, teams composed of less than nine people (54%) are more common in our data set, but we also have teams with more than 20 people (24%) and between 10 and 20 people (22%).

Regarding the size and age of the systems, the most common system size is 10 to 100 KLOC (35%), followed by systems with 100KLOC to 1MLOC (30%), less than 10KLOC (14%), 1 to 10MLOC (14%), and more than 10MLOC (7%). Systems with two to five years (34%) are more common in our data set. However, we also found systems with 1 to 2 years (23%), less than 1-year-old (17%), 5 to 10 years (15%), and more than ten years (11%).

In further analysis of Figure 2, we observe that most of the participants identified themselves as developers (50%), followed by project leader or manager (17%), software architect (13%), tester (7%), process analyst (3%), and other roles (10%). Regarding the level of experience, the participants also identified themselves as competent (34%) and proficient (31%), but we also have experts (21%), beginners (12%), and novices (2%). Concerning the process models, 45% adopted a hybrid process model (a combination of agile and non-agile methods), 42% agile, and 13% the traditional one.

In summary, our data set comprises several participant's roles and levels of experience, companies of different sizes, and projects of different ages, size, team size, and process models. Thus, the data set represents a very wide variety of software development contexts.

4.2 RQ1: What are the leading practices for monitoring TD items in software projects?

In Q24, 53% of the participants indicated that the TD item described in Q13 was monitored and 75% of those described how TD was monitored in Q25. For answering RQ1 and RQ1.1, we used this subset of responses, comprised of 259 answers.

Table 6. Top ten cited TD monitoring-related practices**.**

TD Monitoring-related practice	**#CMRP**	**%MRPP**
TD item backlog	34	13%
Use of tools	31	12%
Team meetings	23	9%
Improving software development process	20	8%
Refactoring	18	7%
Improving tests	17	7%
Code review	16	6%
Communicating the stakeholders of TD items	16	6%
Tracking TD items	12	5%
TD management plan	11	4%

Caption:
#CMRP - Count of monitoring-related practices.
%MRPP - Percentage of CMRP in relation to the total of all projects (259).

[1] The raw data is available at https://doi.org/10.5281/zenodo.13291367.

We identified 46 TD monitoring-related practices. Table 6 summarizes the ten most commonly cited ones. This table reports the monitoring-related practice name and the total number (i.e., count) of citations (#CMRP). #CMRP also indicates the number of projects that used that practice. Column %MRPP presents the percentage of #CMRP concerning the total of projects, revealing how frequently each practice was used in software projects. All identified practices are presented in Figure 8, and Table 15 in the Appendix presents quotes from the participants for each practice.

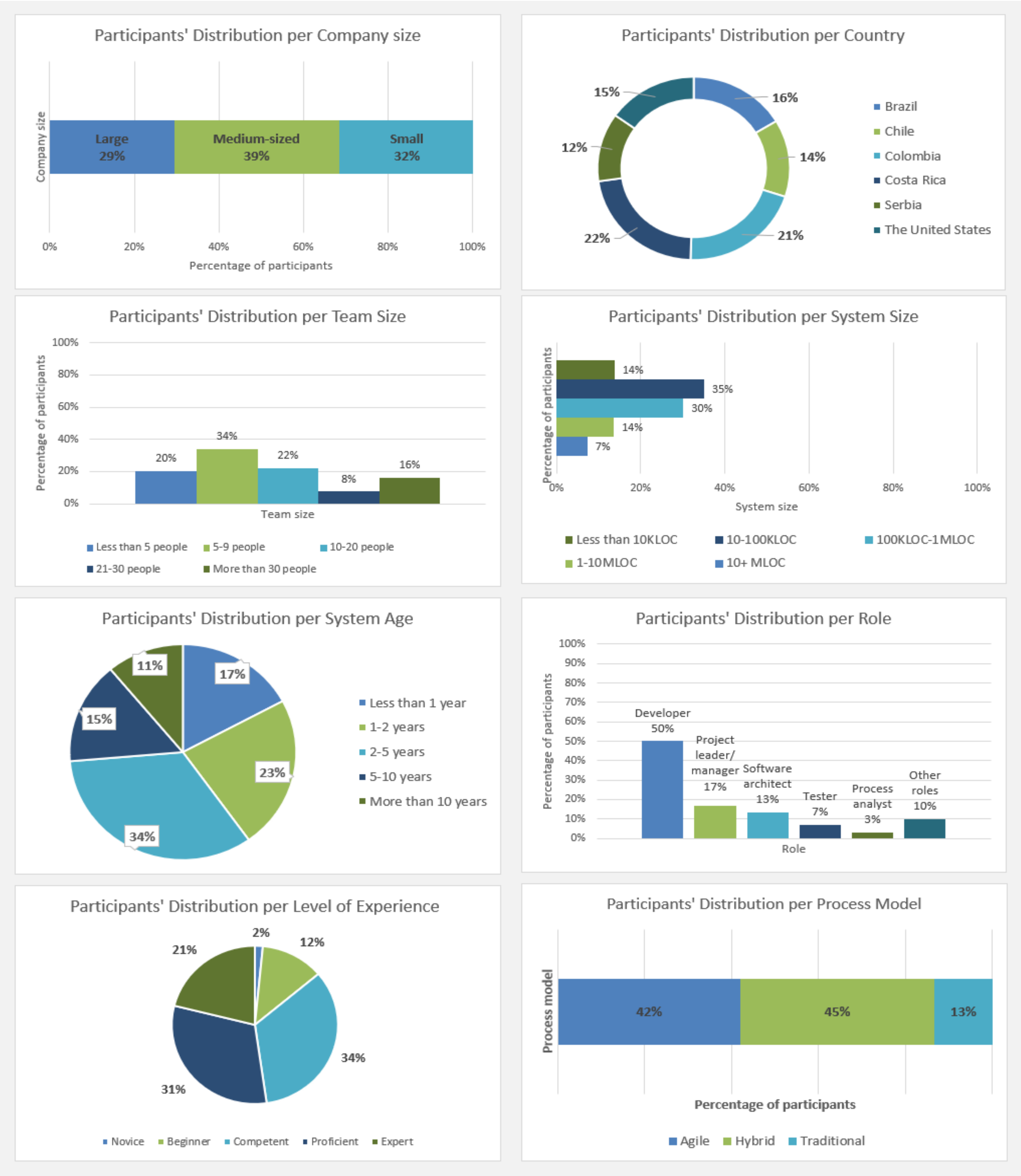

Figure 2. Participants' characterization.

We notice that *TD item backlog* is the most cited practice and has been used in 13% of the projects, followed by *use of tool*, *team meetings*, and *improving software development process* that has been used in 12%, 9%, and 8% of the projects, respectively. In the context of this study, *TD item backlog* means that software teams have included TD items in their task list to prioritize those items and define the most appropriate time for paying them off, as described in the quotes extracted from the answers: "*keep a list of all TD and regularly determine how to eliminate it*" and "*identified as a task and placed in the backlog for later prioritization.*" The practice *use of tools* indicates that TD monitoring activities can be supported by tools to extract metrics from the system, as indicated in the following answers: "*we have to use tool to monitor it, this is an ongoing process*" and "*use of tools like SonarQube.*" *Team meetings* means that software teams have used meetings to track the evolution of the cost and the benefit of TD items over the project, as can be observed in the following answers from the participants: "*regular meeting, usually weekly held that would track the progress*" and "*(...) the use of meetings every day to align the understanding of the development team about the solution that was being developed (...).*" The practice *improving software development process* refers to the improvements applied to the process adopted in the project, as we can observe in "*started with a training process, application of agility and the adoption of the best development practices*" and "*get started with programming best practices.*"

Finding 1: TD is mostly monitored by tracking TD item in the backlog, or by using specialized tools, or by discussing the TD item during the team meetings.

The practices *refactoring* and *improving tests* have been used in 7% of the projects, while *code review* and *communicating the stakeholders of TD items* in 6%. *Refactoring* means changes in the software for improving its internal quality, as we can notice in "*refactoring took place before the development of other services took place*" and "*refactoring when appropriate and time allowed.*" *Improving tests* refers to apply more tests in the system or add other testing activities, as we can observe in "*a test plan was made in conjunction with the user and they were run again*" and "*(...) we began to incorporate more end-to-end tests to gate delivery.*" *Code review* means the checking of source code for identifying mistakes or improvement points, "*every pull request changing the login page must be looked at by least 2 architects*" and "*(...) code reviews done during sprint.*" *Communicating the stakeholders of TD items* means that stakeholders are aware of the TD items existing in the project, as we can observe in "*the client is perfectly aware of it*" and "*it was monitored in the sense that the team had a shared awareness and discussed its ongoing impact.*"

Lastly, the practices *tracking TD items* and *TD management plan* have been used in 5% and 4% of the projects, respectively. *Tracking TD items* refer to providing a view of all TD items existing in the project, as we can notice in "*work considered part of the technical debt was tracked*" and "*through the task/issue tracking system.*" *TD management plan* aims to define a plan to deal with TD items, as we can observe in "*a schedule was developed specifically to address the case*" and "*we were aware of what we are doing and planned to "pay the debt" later on in maintenance stage of the project.*"

Taking a look at the top ten monitoring practices, we noticed that *TD item backlog*, *team meetings*, *communicating the stakeholders of TD items*, and *tracking TD items* are practices that allow the direct monitoring of TD items, while the others provide different types of support to TD monitoring initiative. Thus, we identify one dimension of TD monitoring practices, ***type***, having the following values:

- **Monitoring action**: refers to practices directly related to TD monitoring, such as *TD item backlog*, *communicating the stakeholders of TD items*, and *tracking TD items*.
- **Enabling TD monitoring**: includes practices that improve the capacity of development teams to monitoring debt items. Among them, we have *use of tools*, *improving software development process*, and *assign team for TD monitoring*.
- **TD prevention**: refers to practices intended to avoid potential TD items from being incurred. Some examples are *improving tests*, *code review*, and *qualified professionals*.
- **TD identification**: groups practices that support the identification of TD items in the project. Only *identifying TD items* and *use metrics for TD identification* have this type.
- **TD repayment**: includes practices for repaying TD items. We found only the practices *focusing on TD repayment*, *improve documentation*, and *refactoring* within this type.

Table 7 shows the identified types, reporting the type's name, the number of unique monitoring-related practices cited (#MRP), and the total number (i.e., count) of monitoring-related practices (#CMRP) cited in each type. The column #CMRP also indicates the number of projects that used that practice. Column %MRPP corresponds to the percentage of #CMRP in relation to the total of all projects, indicating how frequently each type was used in software projects. *Monitoring action* is the most cited type by the participants, being used in 49% of the projects. The type *enabling TD monitoring* was also commonly cited by the respondents and was used in 38% of the projects. The types *TD prevention*, *TD repayment*, and *TD identification* were used in 18%, 11%, and 5% of the projects.

Table 7. Types of monitoring-related practices.

Type of monitoring-related practice	#MRP	#CMRP	%MRPP*
Monitoring action	17	127	49%
Enabling TD monitoring	15	97	38%
TD prevention	9	47	18%
TD repayment	3	29	11%
TD identification	2	12	5%

Caption:
#MRP - Count of unique cited monitoring-related practices.

#CMRP - Count of monitoring-related practices.
%MRPP - Percentage of CMRP in relation to the total of all projects (259).
* The percentage exceeds 100% due to the overlapping of practices in a same survey's answer.

This result indicates that, besides using *monitoring actions*, software practitioners have applied other types of practices for supporting TD monitoring initiatives. TD monitoring is part of a bigger process that encompasses TD prevention, repayment, identification, and enabling the TD monitoring.

Finding 2: Direct monitoring actions, supported with enabling tools and practices, are the most common way of monitoring TD.

We also grouped the set of monitoring-related practices into seven ***categories*** (see a description of each of the categories in Table 5):

- **Development issues**: groups practices that are applied during the implementation of the software. Only the practice *improve documentation* composes this category.
- **Infrastructure**: includes two practices related to tools, technologies, and development infrastructure. In this category we have *infrastructure monitoring* and *use of tools*.
- **Internal quality issues**: encompasses four practices that can be employed to address limitations that compromise the internal quality of the software, such as *identify the worst debt areas*, *identifying TD items*, *refactoring*, and *understanding the cause of TD item*.
- **Methodology**: refers to 16 practices related to processes followed by a software team. Examples of practices in this category are *improving software development process* and *improving tests*.
- **Organizational**: includes four practices related to organizational decisions. The practices *knowledge sharing*, *qualified professionals*, *team restructuring*, and *training* compose this category.
- **People**: groups two practices (*communicating the stakeholders of TD items* and *team meetings*) related to the members of software development teams.
- **Planning and management**: refer to 17 practices associated with management activities. Among them, we highlight *TD item backlog*, *tracking TD items*, and *TD management plan*.

Table 9 presents the categories, reporting the category's name, the number of unique monitoring-related practices cited (#MRP) and the total number (i.e., count) of monitoring-related practices (#CMRP) cited in each category. We counted repeated practices found in a participant's response as a single count, the column #CMRP also indicates the number of projects that used a practice for monitoring TD items in each category. Lastly, the column %MRPP corresponds to the percentage of #CMRP in relation to the total of all projects. We notice that the categories *planning and management* and *methodology* concentrate the greatest number of practices. Their practices have been used in 44% and 32% of the projects, respectively.

Table 8. Relationship between categories and types of monitoring-related practices.

Category of monitoring-related practice	Type of monitoring-related practice				
	Monitoring action	**Enabling TD monitoring**	**TD repayment**	**TD prevention**	**TD identification**
Planning and management	**11**	4	1	0	1
Methodology	2	**7**	0	**7**	0
People	**2**	0	0	0	0
Infrastructure	0	**2**	0	0	0
Internal quality issues	**2**	0	1	0	1
Organizational	0	**2**	0	**2**	0
Development issues	0	0	**1**	0	0

Table 9. Categories of monitoring-related practices.

Category of monitoring-related practice	#MRP	#CMRP	%MRPP*
Planning and management	17	115	44%
Methodology	16	84	32%
People	2	39	15%
Infrastructure	2	32	12%
Internal quality issues	4	27	10%
Organizational	4	9	3%
Development issues	1	6	2%

Caption:
#MRP - Count of unique cited monitoring-related practices.
#CMRP - Count of monitoring-related practices.
%MRPP - Percentage of CMRP in relation to the total of all projects (259).
* The percentage exceeds 100% due to the overlapping of practices in a same survey's answer.

Finding 3: Majority of TD monitoring practices represent a dedicated management tasks or a dedicated activity in an overall development process.

Lastly, we analyze the relationship between the categories and types of monitoring-related practices. Table 8 presents the number of practices by categories and types, reporting the category's name, the type's name, and the number of monitoring-related practices by type and category. We notice that the category *planning and management* did not have practices for TD prevention and concentrates the greatest number of monitoring actions. The categories *people* and *internal quality issues* mainly concentrate on monitoring actions. The categories *methodology* and *organizational* have more practices for enabling TD monitoring and TD prevention, while the category *infrastructure* concentrates practices for enabling TD monitoring. Finally, the category *development issues* only has one practice for TD repayment. All identified relationships can support software teams in choosing practices according to teams' needs. For example, if a software team intends to enable TD monitoring, the team can start applying practices from the category *methodology*. However, if the team can implement monitoring actions, it can use practices from the category *planning and management* and also considering practices from the categories: *methodology, people,* and *internal quality issues*.

4.2.1 RQ1.1: What are the practices used for monitoring each type of debt?

Figure 3 presents the relationship among the types of debt and the top 10 TD monitoring-related practices. All practices from the top 10 were applied to monitor code and design debt items. For architecture and test debt, only the practices *code review* and *improving software development process* were not applied, respectively. The practices *improving tests*, *communicating the stakeholders of TD items*, and *TD management plan* were not applied for supporting the monitoring of requirements debt. Documentation debt items were monitored using five practices, while software practitioners have used four practices for people, defect, and process debt. Infrastructure debt was monitored by using three practices. Lastly, build, service, usability, and versioning debt items were only monitored using two practices.

The practice *TD item backlog* is the most used practice for monitoring architecture, build, defect, service, usability, and versioning debt items. Further, *use of tools* is more used for enabling the monitoring of architecture, code, and design debt items. Requirements and process debt items are commonly monitored by *team meetings*, while *improving software development process* and *tracking TD items* are used for monitoring documentation and infrastructure debt items, respectively. Lastly, people and test debt items are commonly monitored by *communicating the stakeholders of TD items* and *improving tests*.

Finding 4: Most pervasive TD monitoring practice is tracking TD items in backlog since this practice can be used for monitoring 12 out of 14 TD types. The practices that follow are the use of specialized tools and code refactoring.

We also investigated the relationship between types of TD and the categories of TD monitoring-related practices. Figure 4 shows this relationship, indicating that the category *planning and management* encompasses practices used for monitoring almost all types of debt, except people and process debt items. Practices from the category *methodology* are not used only for monitoring build, infrastructure, and usability debt items, while documentation, service, usability, and versioning debt items are not monitored by practices from the category *internal quality issues*.

Finding 5: Planning and management related practices are dominantly used for monitoring all types of debt except for people and process debt where they are not used at all.

4.3 RQ2: What are the leading practice avoidance reasons to explain the non-monitoring of TD items?

In total, 47% of the participants did not monitor the TD item described in Q13. Of them, 64% explained in Q25 why the TD was not monitored. This set of responses (197) was used to answer RQ2 and RQ2.1.

We identified 35 PARs for TD non-monitoring. Table 10 presents the ten most commonly cited ones. This table reports the PAR name and the total number (i.e., count) of citations (#CPAR). #CPAR also indicates the number of projects that used a PAR for justifying the TD non-monitoring. Column %PARP presents the percentage of #CPAR to the total of all projects, revealing how frequently each PAR was considered in software projects. All identified PARs for TD non-monitoring are presented in Figure 8, and Table 16 in the Appendix presents quotes from the participants for each PAR.

Table 10. Top ten cited PARs for TD non-monitoring.

Practice avoidance reason (PAR)	#CPAR	%PARP
Lack of interest	44	22%
Focusing on short term goals	33	17%
Lack of time	29	15%
Lack of knowledge on TD	23	12%
Lack of understanding about the impact of the debt	12	6%
Lack of organizational culture	8	4%
Lack of resources	8	4%
Lack of TD monitoring process	7	4%
Lack of specific team	6	3%
React when becoming a problem	5	2%

Caption:
#CPAR - Count of practice avoidance reason for TD non-monitoring.
%PARP - Percentage of CPAR in relation to the total of all projects (197).

Lack of interest is the most cited PAR, being considered in 22% of the projects. The other best-positioned PARs are *focusing on short term goals*, *lack of time*, and *lack of knowledge on TD,* which were considered in 17%, 15%, 12% of the projects, respectively. *Lack of interest* refers to organizations or project managers that do not purse to monitor TD items, as indicated in the following answers:

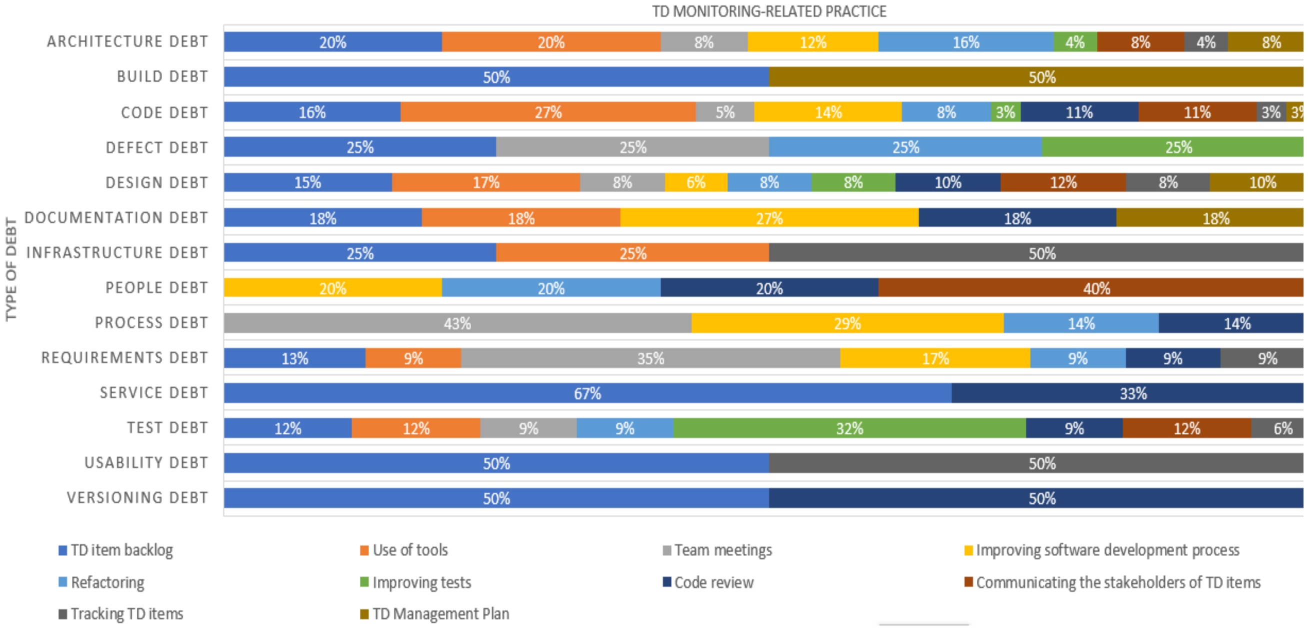

Figure 3. Relationship among types of debt and the top 10 TD monitoring-related practices.

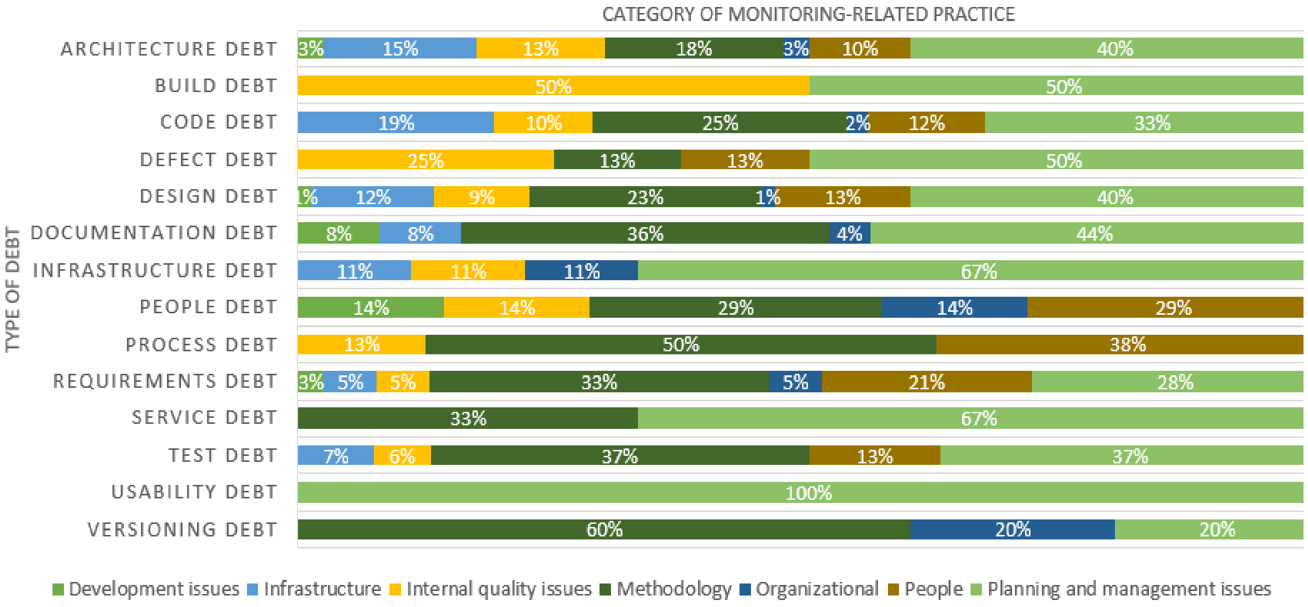

Figure 4. Relationship among types of debt and categories of TD monitoring-related practices.

"*little interest of the company to correct this type of situation*" and "*management did not care.*" The PAR *focusing on short term goals* means that software teams have other priorities in their projects, as informed in "*not in the priority pipeline*" and "*it was not critical for the success of the project.*" *Lack of time* refers to software projects that do not have time for TD monitoring, as we can notice in "*because deadlines are tight*" and "*the project timeline didn't allow it.*" The PAR *lack of knowledge on TD* means that software teams do not monitor debt items because they did not have knowledge on TD, such as we can observe in the answers "*because the concept of technical debt was not yet applied in the company*" and "*there is no knowledge about TD.*"

Finding 6: *Lack of interest*, *lack of time*, and the *focus on short term goal* are the main reasons why companies avoid monitoring TD items.

The PAR *lack of understanding about the impact of the debt* was considered in 6% of the projects, while *lack of organizational culture*, *lack of resources*, and *lack of TD monitoring process* were considered, each one, in 4% of the projects. *Lack of understanding about the impact of the debt* indicates that although software teams have identified TD items in their projects, they did not become aware of how much those items can impact those projects: "*lack of knowledge of the impact of the TD item in question*" and "*technical debt was not identified as a problem at the time.*" The PAR *lack of organizational culture* refers to organizations that do not consider TD management as part of their development activities: "*this is very dependent on the organization. In this case, it was not identified, monitored, or managed*" and "*because there is no permanent initiative to generate changes in the organizational culture.*" *Lack of resources* makes unfeasible the monitoring of TD items, as we can notice in "*even knowing the problem, there are no resources for immediate solution*" and "*the time and resources of the project were very limited.*" The PAR *lack of TD monitoring process* means that a process to monitor TD is missing in the organization, as evidenced in "*there was no process for it*" and "*we weren't tracking it.*"

Lastly, the PARs *lack of specific team* and *react when becoming a problem* were considered in 3% and 2% of the projects, respectively. *Lack of specific team* refers to software projects that do not have a team responsible for monitoring TD: "*it was known about for years but we didn't have the headcount to refactor*" and "*not enough people or time too.*" *React when becoming a problem* indicates the existence of bad practices when managing with TD, as we can observe in "*it is neglected until it becomes a problem*" and "*in the absence of planning, the methodology was reactionary to the problems.*"

As with monitoring practices, the data revealed distinct ***types*** of PARs. After analyzing the top ten PARs, we noticed that items like *lack of interest*, *focusing on short term goals*, and *react when becoming a problem* represent a **decision factor** considered by the team for not monitoring TD items. Differently, *lack of time* and *lack of knowledge on TD* represent situations in which the practitioners could have the intention of monitoring the debt, but they would not be able to due to issues that are out of their control (an **impediment**).

Finding 7: TD is not monitored due to explicit decisions not to monitor the debt or due to impediments that obstruct teams or team members in monitoring.

Table 11 presents the identified types of PARs, reporting the type's name, the number of unique PARs for TD non-monitoring cited (#PAR), and the total number (i.e., count) of PARs (#CPAR) cited in each type. #CPAR also indicates the number of projects that used that PAR for justifying the non-monitoring of TD items in each type. Column %PARP corresponds to the percentage of #CPAR in relation to the total of all projects, indicating how frequently each type was used in software projects. *Impediments* are the most common reason for explaining the non-monitoring of TD, being present in 64% of the software projects, while *decision factors* were considered in 50% of the projects.

Table 11. Type of PAR for TD non-monitoring.

Type of PAR for TD non-monitoring	#PAR	#CPAR	%PARP*
Decision factor	10	98	50%
Impediment	25	125	64%

Caption:
#PAR - Count of unique cited practice avoidance reason (PAR) for TD non-monitoring.
#CPAR - Count of practice avoidance reasons for TD non-monitoring.
%PARP - Percentage of CPAR in relation to the total of all projects (197).
* The percentage exceeds 100% due to the overlapping of PARs in a same survey's answer.

When analyzing the identified set of PARs, we noticed that many of them were associated with each other and could be organized into more generic ***categories***. For example, the PARs *lack of TD monitoring process* and *TD item eliminated as soon as identified* could be grouped into a category like *methodology*, because they are related to the process adopted by a software team. To name the categories, we used the categories proposed by Rios *et al*. (2020b), as presented in Table 5. Then, we grouped the PARs into eight categories:

- **Development issues**: encompasses PARs associated with software development activities. The two PARs grouped in this category are: *changing in the requirements* and *legacy system*.
- **External factors**: refers to PARs associated with factors that software teams cannot control. The three identified PARs in this category are *business pressure*, *project discontinued*, and *TD item payment do not generate revenue*.
- **Internal quality issues**: groups PARs associated with limitations that compromise the internal quality of the software. The PARs *complexity of TD items*, *lack of effort to know the cause of TD*, and *too many TD items* compose this category.
- **Lack of knowledge**: includes only one PAR (*lack of knowledge on TD*) associated with the need for technical knowledge.
- **Methodology**: encompasses PARs related to processes followed by a software team. Examples of PARs in this category are *lack of TD monitoring process*, *react when becoming a problem*, and *TD item eliminated as soon as identified*.
- **Organizational**: includes PARs related to organizational decisions. Among them, we highlight *lack of interest*, *lack of organizational culture*, and *lack of resources*.
- **People**: refers to PARs associated with team characteristics, such as *emotional issues of the team*, *lack of experience*, and *team overload*.
- **Planning and management**: groups PARs related to management activities. Examples are *focusing on short term goals*, *lack of time*, and *ineffective planning and management*.

Table 12 presents the categories of PARs for TD non-monitoring, reporting the category's name, the number of unique PARs cited (#PAR) and the total number (i.e., count) of PARs (#CPAR) cited in each category. #CPAR also indicates the number of projects that considered a PAR for explaining the non-monitoring of TD items in each category. Lastly, the column %PARP corresponds to the percentage of #CPAR in relation to the total of all projects.

The categories *planning and management* and *organizational* have the greatest number of PARs, impacting 46% and 36% of the projects, respectively. One could assume that technical (development and internal quality) issues would have a decisive role when opting for not monitoring debt items. However, contrary to this, our results pointed out that managerial aspects are the key concern we should have in mind to understand why the monitoring of debt items has not been considered in the projects.

Finding 8: Most of the reasons why companies refuse to monitor TD items originate from management, but also from organization as an overall working context.

Table 12. Categories of PARs for TD non-monitoring.

Category of PAR for TD non-monitoring	#PAR	#CPAR	%PARP*
Planning and management	10	90	46%
Organizational	6	70	36%
Lack of knowledge	1	23	12%
Methodology	6	20	10%
People	4	7	4%
External factors	3	5	3%
Development issues	2	4	2%
Internal quality issues	3	4	2%

Caption:
#PAR - Count of unique cited practice avoidance reasons (PARs) for TD non-monitoring.
#CPAR - Count of PARs for TD non-monitoring.
%PARP - Percentage of CPAR in relation to the total of all projects (197).
* The percentage exceeds 100% due to the overlapping of PARs in a same survey's answer.

Lastly, we investigated the relationship between the categories and types of PARs, as shown in Figure 5. The categories *planning and management*, *organizational*, and *people* have the greatest number of impediments, revealing that these categories deserve more attention if we want to increase the capacity of a team's investment on TD monitoring activities. This is also true for the categories *lack of knowledge*, *external factors*, and *development issues*. On the other hand, *methodology* and *internal quality issues* are more related to decision factors. This is expected, because these two categories are closer to daily and recurrent activities directly performed by teams. An investment in facilitating the execution of TD monitoring activities at this level can have an impact on the management of TD items.

4.3.1 RQ2.1: What are the main practice avoidance reasons for non-monitoring each type of debt?

Figure 6 presents the relationship among the types of debt and the top 10 PARs. The PARs *lack of time* and *lack of knowledge on TD* are considered to explain the non-monitoring of most TD types; then, software teams can facilitate TD monitoring improving their planning and spending time in training on TD. We can also observe that the explanation of the non-monitoring of service, usability, and versioning debt items are only done by the PARs *lack of knowledge on TD*, *focusing on short term goals*, and *lack of interest*, revealing that besides promoting their knowledge on TD, software teams need to change their and their organization's mindset. Lastly, as test and documentation debt items share almost the same PARs, teams wanting to monitor these items can apply some strategies for facilitating it.

Finding 9: Most of TD types (13 of 14) are not monitored due to at least one of the following reasons: *lack of interest*, *lack of time*, or *focusing on short term goals*.

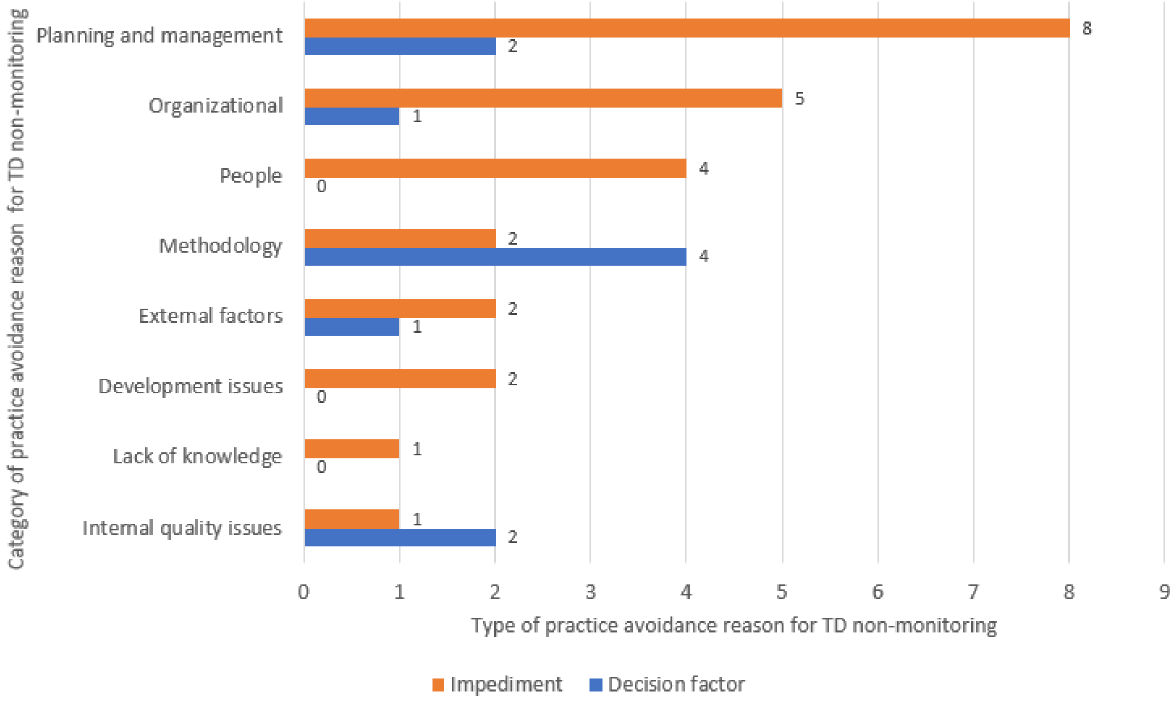

Figure 5. Relationship among categories and types of PARs for TD non-monitoring.

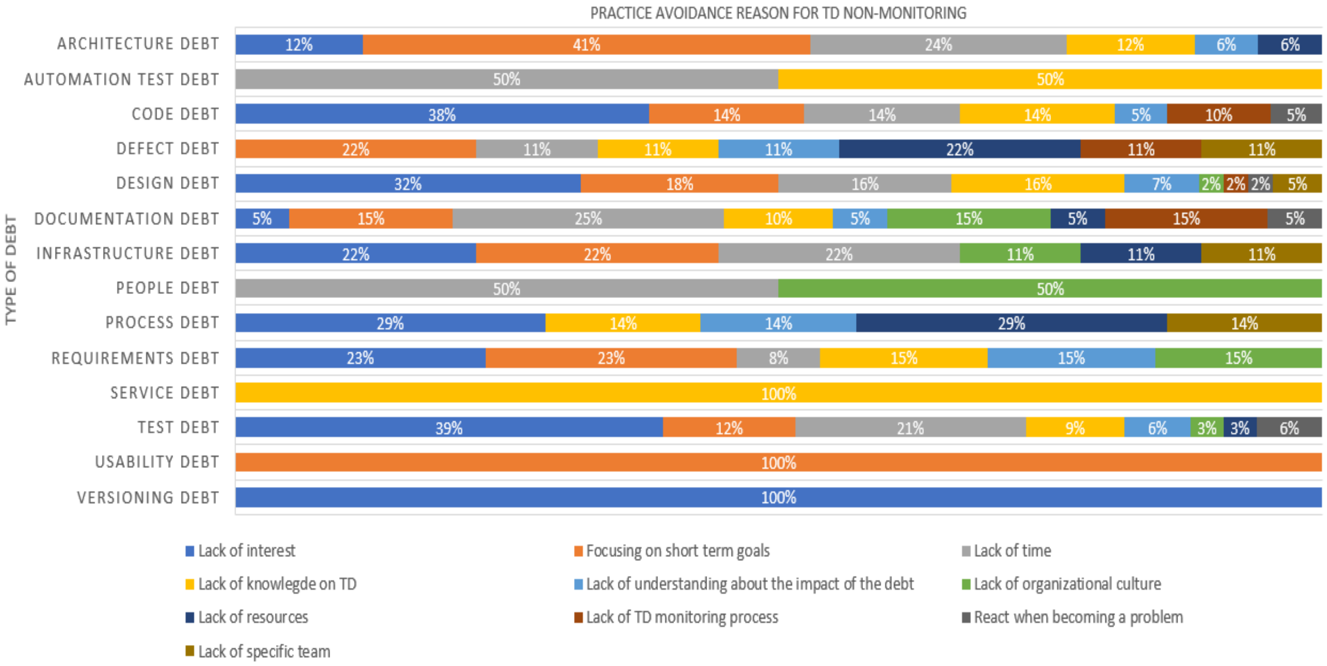

Figure 6. Relationship among types of debt and the top 10 TD monitoring-related practices.

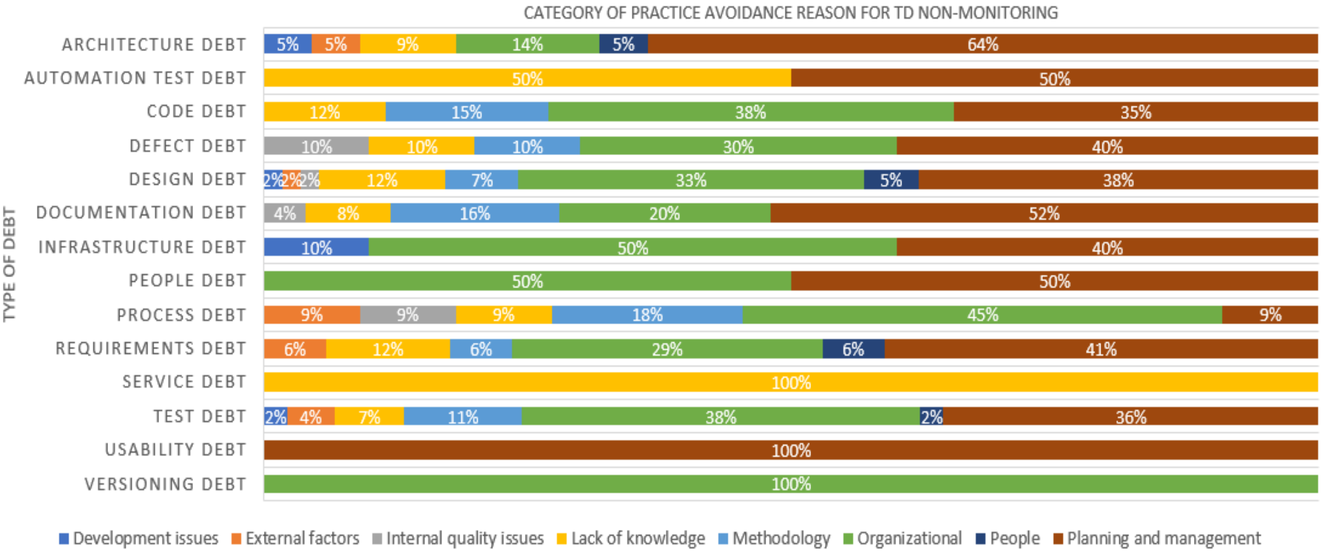

Figure 7. Relationship among types of debt and categories of TD monitoring-related practices.

Figure 7 shows the relationship among the types of debt and categories of PARs, indicating how frequently PARs from these categories are used to explain the non-monitoring of each type of debt. The categories *planning and management* and *organizational* concentrate the PARs of almost all types of debt, indicating that software teams can improve their capacity for TD monitoring investing efforts to promote their managerial activities and organizational decisions. Although PARs from those categories are mainly used for explaining the non-monitoring of architecture, code, defect, design, documentation, infrastructure, people, process, requirements, test, usability, and versioning debt items, we noticed that the non-monitoring of automation test and service debt items are commonly justified by PARs from the category *lack of knowledge*. It means that software teams can improve their technical knowledge for enabling TD monitoring initiatives.

5 Discussion

In this section, we revisit the findings of each research question. After, we organize the TD monitoring-related practices and PARs into a conceptual map.

5.1 Revisiting the research questions

We identified 46 monitoring-related practices for monitoring TD items (RQ1). Of these practices, *TD item backlog*, *use of tools*, *team meetings*, and *improving software development process* are the most used. The identified practices have been used by practitioners for monitoring, preventing, identifying, and repaying TD items, and enabling TD monitoring initiatives. It reveals that efforts for monitoring TD items should take into consideration the integration among these TD management activities. Also, we grouped the practices into seven categories. *Planning and management* and *methodology* encompass the majority of the practices, playing a central role in TD monitoring. Practices from both categories benefit from each other, once the category *planning and management* is mainly characterized for grouping *monitoring actions*, while *methodology* has more practices for *enabling TD monitoring*. Lastly, the ten most cited practices are performed for monitoring the analyzed 14 types of debt. Practices from *planning and management* are used for monitoring almost all analyzed types, indicating that it would be a good choice to start TD monitoring initiatives by them.

Concerning the PARs for explaining the non-monitoring of TD items (RQ2), we identified 35 of them. *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. In general, impediments are more commonly considered for justifying the non-monitoring of TD items than decision factors. Thus, the monitoring of TD is not just a matter of will, but of mitigating the restrictions that curb the adoption of monitoring practices too.

We grouped the PARs into eight categories. *Planning and management* and *organizational* concentrate the greatest number of PARs, revealing that decisions taken at organizational or managerial levels are decisive for monitoring TD items. Finally, we investigated the relation between PARs and types of debt. Overall, the ten most cited PARs have been considered for justifying the non-monitoring of all types of debt. Particularly, PARs from *planning and management* and *organizational* have been considered for explaining the majority of the analyzed types.

5.2 Technical debt monitoring conceptual map

In this section, we organize the TD monitoring-related practices and PARs for TD non-monitoring into a conceptual map, following the concepts of evidence briefings (Cartaxo *et al*. 2016). Figure 8 shows the conceptual map for TD monitoring. 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. To calculate the percentages, we summed up the number of occurrences for each practice or PAR and divided by the number of projects in which that practice or PAR was cited. For example, the monitoring action *TD item backlog* was cited by 34 participants. As we had 259 participants indicating that a TD item was monitored in their projects, *TD item backlog* was used in 13% (34/259*100) of them. Finally, to calculate the percentage of each category, we summed up the percentages of its practices or PARs.

We use small circles with different colors for representing the types of TD monitoring-related practices (monitoring action, enabling TD monitoring, TD prevention, TD payment, or TD identification) and PARs (decision factor or impediment). For example, the category *internal quality issues*, from the monitoring-related practices rectangle, has one white circle (*identifying TD items*) representing a practice of the *TD identification* type, one brown circle (*refactoring*) of the *TD repayment* type, and two gray circles (*understanding the cause of TD item* and *identify the worst debt areas*) of the *monitoring action* type.

The map shows us, for example, that the categories *planning and management* and *methodology* are the most commonly used in 44% and 32% of the projects, respectively. From the category *planning and management*, the monitoring action *TD item backlog* stands out, being used in 13% of the projects. On the right side, the map indicates that the categories *planning and management* (46%) and *organizational* (36%) are the most commonly considered PARs for explaining the non-monitoring of TD items. The decision factor *lack of interest* stands out in the category *organizational,* is considered 22% of the projects.

Concerning the practices, the map can be useful for practitioners in two scenarios. If a team does not have experience performing TD monitoring, it can base its first steps on the experience of others and use the percentages as a criterion for choosing practices to set its monitoring initiative. In the second scenario, if a team already has experience TD monitoring, the map serves as a benchmarking tool. Based on the experience from others, the team can compare its practices and identify new ones that could be used. In addition, when looking at the categories' level, if a team already use a practice from a specific category, it can discover other practices related to that practice.

About the PARs, the map sheds light on possible improvement points in the team's capability to make feasible the application of practices for monitoring the debt. These points can be divided into two scenarios. First, let us consider a team with no experience in monitoring TD items, the map can support the team in identifying PARs used in practice, and the percentages can be used as a criterion for verifying what PARs are more common for impeding TD monitoring initiatives. Lastly, if the team has already experienced TD monitoring, the map can reveal new PARs from a practitioners' experience, improving the team's perception of factors that curb TD monitoring. Also, the map categories can support the team in identifying other PARs related to ones already used by the team associated with the same TD monitoring issue. In both scenarios, as PARs are divided into impediments and decision factors, software teams can understand whether the TD non-monitoring occurs due to their decision, or an impediment posed by other stakeholders.

5.3 Comparison to Related Work

We compared our findings with the ones reported in related work. As previously said, we did not find studies that investigated the PARs, thus, the comparison only takes into consideration the practices used for monitoring TD items. The comparison was performed in two levels: practices and categories of practices. To perform this comparison, the first author identified the practices or categories in related work and mapped them to corresponding ones found in our work. Afterwards, the last author reviewed this comparison, and the divergences were resolved in a meeting involving these authors.

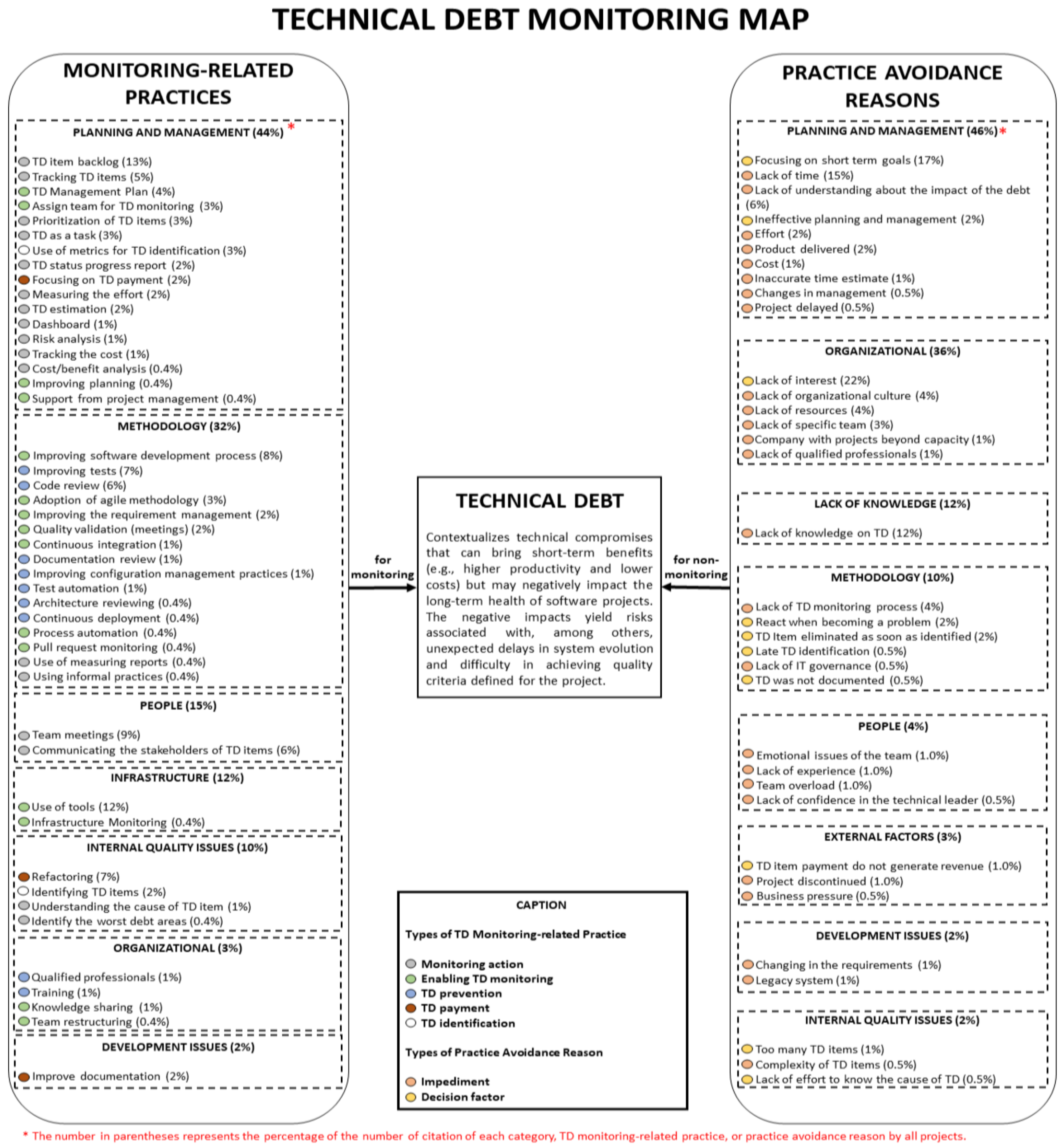
TECHNICAL DEBT MONITORING MAP
MONITORING-RELATED PRACTICES
PLANNING AND MANAGEMENT (44%) *
TD item backlog (13%)
Tracking TD items (5%)
TD Management Plan (4%)
Assign team for TD monitoring (3%)
Prioritization of TD items (3%)
TD as a task (3%)
Use of metrics for TD identification (3%)
TD status progress report (2%)
Focusing on TD payment (2%)
Measuring the effort (2%)
TD estimation (2%)
Dashboard (1%)
Risk analysis (1%)
Tracking the cost (1%)
Cost/benefit analysis (0.4%)
Improving planning (0.4%)
Support from project management (0.4%)
METHODOLOGY (32%)
Improving software development process (8%)
Improving tests (7%)
Code review (6%)
Adoption of agile methodology (3%)
Improving the requirement management (2%)
Quality validation (meetings) (2%)
Continuous integration (1%)
Documentation review (1%)
Improving configuration management practices (1%)
Test automation (1%)
Architecture reviewing (0.4%)
Continuous deployment (0.4%)
Process automation (0.4%)
Pull request monitoring (0.4%)
Use of measuring reports (0.4%)
Using informal practices (0.4%)
PEOPLE (15%)
Team meetings (9%)
Communicating the stakeholders of TD items (6%)
INFRASTRUCTURE (12%)
Use of tools (12%)
Infrastructure Monitoring (0.4%)
INTERNAL QUALITY ISSUES (10%)
Refactoring (7%)
Identifying TD items (2%)
Understanding the cause of TD item (1%)
Identify the worst debt areas (0.4%)
ORGANIZATIONAL (3%)
Qualified professionals (1%)
Training (1%)
Knowledge sharing (1%)
Team restructuring (0.4%)
DEVELOPMENT ISSUES (2%)
Improve documentation (2%)
for monitoring
TECHNICAL DEBT
Contextualizes technical compromises that can bring short-term benefits (e.g., higher productivity and lower costs) but may negatively impact the long-term health of software projects. The negative impacts yield risks associated with, among others, unexpected delays in system evolution and difficulty in achieving quality criteria defined for the project.
for non-monitoring
CAPTION
Types of TD Monitoring-related Practice
Monitoring action
Enabling TD monitoring
TD prevention
TD payment
TD identification
Types of Practice Avoidance Reason
Impediment
Decision factor
PRACTICE AVOIDANCE REASONS
PLANNING AND MANAGEMENT (46%)*
Focusing on short term goals (17%)
Lack of time (15%)
Lack of understanding about the impact of the debt (6%)
Ineffective planning and management (2%)
Effort (2%)
Product delivered (2%)
Cost (1%)
Inaccurate time estimate (1%)
Changes in management (0.5%)
Project delayed (0.5%)
ORGANIZATIONAL (36%)
Lack of interest (22%)
Lack of organizational culture (4%)
Lack of resources (4%)
Lack of specific team (3%)
Company with projects beyond capacity (1%)
Lack of qualified professionals (1%)
LACK OF KNOWLEDGE (12%)
Lack of knowledge on TD (12%)
METHODOLOGY (10%)
Lack of TD monitoring process (4%)
React when becoming a problem (2%)
TD Item eliminated as soon as identified (2%)
Late TD identification (0.5%)
Lack of IT governance (0.5%)
TD was not documented (0.5%)
PEOPLE (4%)
Emotional issues of the team (1.0%)
Lack of experience (1.0%)
Team overload (1.0%)
Lack of confidence in the technical leader (0.5%)
EXTERNAL FACTORS (3%)
TD item payment do not generate revenue (1.0%)
Project discontinued (1.0%)
Business pressure (0.5%)
DEVELOPMENT ISSUES (2%)
Changing in the requirements (1%)
Legacy system (1%)
INTERNAL QUALITY ISSUES (2%)
Too many TD items (1%)
Complexity of TD items (0.5%)
Lack of effort to know the cause of TD (0.5%)
* The number in parentheses represents the percentage of the number of citation of each category, TD monitoring-related practice, or practice avoidance reason by all projects.

Figure 8. TD monitoring map.

Table 13 presents the comparison considering the ten best-positioned practices reported in our study. Analyzing the table, we can observe that half of these practices are confirmed in at least one related work, indicating that the results are somewhat aligned. The most cited practices *TD item backlog* and *use of tools* are previously found by four and six related work. It evidences that different types of research (case studies, tool development, systematic review, and survey) also provided a cohesive list of practices for TD monitoring. In summary, we found 19 practices standard for both studies (ours and related work), but other 27 practices were found only in our study, and only six practices were recognized in the related work. The complete comparison is presented in Table 17 in the Appendix.

Table 14 presents the comparison considering the categories of practices. This comparison was only done with Li *et al.* (2015) because it reported categories of practices for monitoring TD items. The result of a comparison can be total (indicating that all practices from a category identified in our work are considered in the related work and vice-versa) or partial (indicating that our work has practices that are not considered in the related work). We notice that the categories *methodology*, *internal quality issues*, and *planning and management* were confirmed partially in the literature, while the categories *development issues*, *infrastructure*, *organizational*, and *people* were found only in our study.

In summary, our work extends the state of the art providing evidence on the practices used by software practitioners to monitor TD items or support their TD monitoring initiatives. Also, we evidence the primary PARs that hamper the application of these practices, supporting software practitioners in analyzing their projects and define strategies to resolve these PARs.

Table 13. Comparison between the best positioned TD monitoring-related practices and the ones reported by related work.

Our Study	**Technical Debt Monitoring-Related Practices from**										
	RW1	**RW2**	**RW3**	**RW4**	**RW5**	**RW6**	**RW7**	**RW8**	**RW9**	**RW10**	**RW11**
TD item backlog	-	Backlog grooming	-	-	-	Creation of TD items in a backlog	Reporting TD items in backlog	-	Panels (TO DO, DOING, and DONE), based on the Kanban concept	-	-
Use of tools	-	-	-	Using data collected from (management or TD measuring) tools	-	AnaConDebt	Using system for bug fixing	Measuring symptom severity on a smell thermometer Sonar TD plugin DebtFlag TD evaluation (SQALE) Software maps tool Code Christmas tree	VisminerTD	tools	-
Team meetings	-	-	-	-	-	-	-	-	-	-	-
Improving software development process	-	-	-	-	-	-	-	-	-	-	-
Refactoring	-	-	-	-	-	-	-	-	-	-	-
Improving tests	-	-	-	-	-	-	Measuring test coverage	-	-	-	Changes in the test process
Code review	-	-	-	-	-	-	-	SQALE method	-	-	-
Communicating the stakeholders of TD items	-	-	-	-	-	-	-	-	-	-	-
Tracking TD items	-	-	-	-	-	-	-	-	-	-	-
TD Management Plan	Accounting	-	-	-	Planning in advance for TD	-	-	Accounting formal approach to TD decision making	-	-	-

Caption:
RW1: Ampatzoglou *et al.* (2015)
RW2: Ernst *et al.* (2015)
RW3: Oliveira *et al.* (2015)
RW4: Yli-Huumo *et al.* (2016)
RW5: Behutiye *et al.* (2017)
RW6: Martini (2018)
RW7: Martini *et al.* (2018)
RW8: Rios *et al.* (2018b)
RW9: Mendes *et al.* (2019)
RW10: Apa *et al.* (2020)
RW11: Aragão *et al.* (2021)

Table 14. Comparison to related work on categories of TD monitoring-related practices.

Our Categories	Categories from Li *et al.* (2015)		Overlapping Degree
	Category Name	Definition	
Methodology	TD propagation tracking	Track the influences of TD through dependencies between other parts of a system and the parts of the system that contains TD.	Partial
	TD monitoring with quality attribute focus	Monitor the change of quality attributes that detrimental to TD, such as stability.	Partial
	Planned check	Regularly measure identified TD and track the change of the TD.	Partial
Internal quality issues	Threshold-based approach	Define thresholds for TD related quality metrics, and issue warnings if the thresholds are not met.	Partial
Planning and management issues	TD plot	Plot various aggregated measures of TD over time and look at the shape of the curve to observe the trends.	Partial
Development issues	-	-	-
Infrastructure	-	-	-
Organizational	-	-	-
People	-	-	-

5.3.1 Comparing the identified practices and PARs with those reported by Freire *et al.* (2021c)

By analyzing 274 answers given by agile software practitioners, our previous study (Freire *et al.* 2021c) identified 25 monitoring-related practices and 22 PARs for TD non-monitoring. These practices were also recognized in the present study, but we increased the list to 46 practices and 35 PARs.

Regarding the practices, we found the following new ones: *architecture reviewing*, *code review*, *continuous deployment*, *cost/benefit analysis*, *documentation review*, *focusing on TD payment*, *identifying TD items*, *improve documentation*, *improving configuration management practices*, *improving planning*, *improving tests*, *qualified professionals*, *quality validation (meetings)*, *refactoring*, *risk analysis*, *support from project management*, *team restructuring*, *test automation*, *training*, *use of metrics for TD identification*, and *using informal practices*.

About the PARs, we identified the following new ones: *changes in management*, *company with projects beyond capacity*, *complexity of TD items*, *lack of confidence in the technical leader*, *lack of effort to know the cause of TD*, *lack of qualified professionals*, *lack of TI governance*, *late TD identification*, *project delayed*, *project discontinued*, *TD item payment do not generate revenue*, *TD was not documented*, and *team overload*.

Our study confirms and extends the list of practices and PARs reported in our previous study (Freire *et al.* 2021c) based on the increased diversity of software development contexts.

6 Threats to Validity

As in any empirical study, our study can be affected by threats to validity, requiring strategies to remove or mitigate these threats. We identified threats affecting construct, conclusion, internal, and external validity, following the categories defined by Wohlin *et al.* (2012).

Construct validity. Threats from this category are associated with social factors or experimental design (Wohlin *et al.* 2012). We identified a threat arising from the validity of participants' responses. Participants could answer the survey questions without considering the context of TD or TD monitoring. And, as the survey was performed remotely, it can maximize the effect of this threat. To mitigate it, we included two acceptance criteria: (i) the example of TD provided by participants in Q13 must describe an actual TD item and (ii) the answer given to Q25 must be associated with practices or PARs for TD monitoring. Then, a participant's answer was considered in our analysis, if the answer fit into these acceptance criteria.

Conclusion validity. It affects the capacity to correctly interpret the results (Wohlin *et al.* 2012). A threat arises from the qualitative analysis because it is subjective and subject to inconsistencies. We used this analysis type in three activities: identifying TD types from Q13, coding Q25 for recognizing practices and PARs, and grouping these practices and PARs into types and categories. To reduce this threat, two different researchers in each *InsighTD* replication performed the TD type identification separately, considering answers given to Q13. To solve eventual divergences, a third researcher also analyzed the answers. In the coding process, at least three researchers in each *InsighTD* replication identified practices and PARs from answers given to Q25. These researchers performed one of the following roles: code identifier, code reviewer, and referee. Lastly, for grouping practices and PARs into types and categories, a researcher performed the categorization and defined the type, and an experienced researcher reviewed them. Eventual divergences were resolved in a meeting between these researchers.

Internal validity. Threats related to other factors affecting the results without the researcher's knowledge compose this category (Wohlin *et al*. 2012). As the survey questions were answered remotely, the participants could misunderstand these questions, arising an internal threat affect our study. To minimize it, the survey passed through three internal reviews conducted by experienced researchers from the *InsighTD* project, and one external review conducted by a senior researcher. Afterwards, a pilot study was run to assess the survey questions, structure, and duration. More details on this process are described in Rios *et al*. (2020b).

External validity. Regarding the threats that affect our ability to generalize the results, we mitigate them by targeting industry practitioners from different countries, organizations, and projects characteristics. Moreover, the sample (composed of 653 participants) minimizes the chances of subsets of participants have biased the results. However, we are still not able to define how generalizable the results are due to the lack of empirical data characterizing the population. We intend to continuously reduce these threats by collecting and consolidating more empirical evidence from other *InsighTD* replications. However, an argument can be made that the ecological validity (Andrade 2018) of the work, i.e., the extent to which these findings approximate other real-world scenarios, is likely to hold in other settings.

7 Final Remarks

This work reports software practitioners' point of view on TD monitoring, revealing the practices used for monitoring TD and the PARs considered for explaining TD non-monitoring. The work groups practices and PARs into categories, indicating the main issues associated with TD monitoring. It also identifies relationships between practices and types of debt and PARs and types of debt. These relations can support software practitioners in addressing specified practices or PARs to a type of debt that commonly affects their project.

To make our results more useable in practice, we organize practices, PARs, and their categories and types into a conceptual map. Practitioners can use the map as a guideline supporting the identification of TD monitoring practices and PARs. Irrespective of having experience in TD monitoring, software teams can analyze the practices and PARs (i) considering their percentual of occurrence and (ii) discovering practices and PARs related to each other by their categories. Software practitioners can use the conceptual map to support their TD monitoring initiatives. We provide more detail in Section 5.2. Also, we offer the following key takeaways that come from our analysis:

1. TD is mostly monitored by tracking TD item in the backlog, or by using specialized tools, or by discussing the TD item during the team meetings.
2. Direct monitoring actions, supported with enabling tools and practices, are the most common way of monitoring TD.
3. Majority of TD monitoring practices represent a dedicated management tasks or a dedicated activity in an overall development process.
4. Most pervasive TD monitoring practice is tracking TD items in a backlog since this practice can be used for monitoring 12 out of 14 TD types. The practices that follow are the use of specialized tools and code refactoring.
5. Planning and management related practices are dominantly used for monitoring all types of debt except for people and process debt where they are not used at all.
6. *Lack of interest*, *lack of time*, and the *focus on short term goal* are the main reasons why companies avoid monitoring TD items.
7. TD is not monitored due to explicit decisions not to monitor the debt or due to impediments that obstruct teams or team members in monitoring.
8. Most of the reasons why companies refuse to monitor TD items originate from management, but also from organization as an overall working context.
9. Most of TD types (13 of 14) are not monitored due to at least one of the following reasons: *lack of interest*, *lack of time*, or *focusing on short term goals*.

For researchers, our findings can stimulate new research aligned with the TD monitoring state of the practice. The list of practices and PARs can motivate investigations in a problem-driven way. For example, researchers can propose strategies to mitigate the effects of PARs present in software projects by conducting case studies in software industry. Furthermore, the proposal of new strategies and tools for TD monitoring will benefit from combining different TD management strategies with current TD monitoring practices. According to our survey, participants have used practices to monitor TD and for TD repayment, prevention, and identification.

As future work, we intend to empirically assess the conceptual map to verify their effectiveness for supporting TD monitoring. Moreover, we seek to explore the relationship between the effects of TD and TD monitoring to identify the practices or PARs used by a software team when it felt the presence of the debt in the team's project. Finally, the relationship between TD repayment and monitoring could also be investigated to identify how software practitioners integrated the practices used to monitor and repay the debt.

Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) Finance Code 001 and the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

References

Alves, N. S. R., Mendes, T. S., Mendonça, M., Spínola, R., Shull, F., & Seaman, C. (2016). Identification and management of technical debt: a systematic mapping study. *Information and Software Technology*, vol. 70, February, pp. 100-121, doi:10.1016/j.infsof.2015.10.008.

Alzaghoul, E., & Bahsoon, R. (2014). Evaluating technical debt in cloud-based architectures using real options. In Proceedings of *23rd Australian Software Engineering Conference*, Milsons Point, doi: 10.1109/ASWEC.2014.27.

Ampatzoglou, A., Ampatzoglou, A., Chatzigeorgiou, A., & Avgeriou, P. (2015). The financial aspect of managing technical debt: a systematic literature review. *Information and Software Technology*, vol. 64, pp.52-73, doi: 10.1016/j.infsof.2015.04.001.

Andrade, C. (2018). Internal, External, and Ecological Validity in Research Design, Conduct, and Evaluation. *Indian journal of psychological medicine*, 40 (5), 498-499, doi: 10.4103/IJPSYM.IJPSYM_334_18.

Apa, C., Jeronimo, H., Nascimento, L. M., Vallespir, D., & Travassos, G. H.. (2020). The perception and management of technical debt in software startups. In: Nguyen-Duc A., Münch J., Prikladnicki R., Wang X., Abrahamsson P. (eds) *Fundamentals of Software Startups*. Springer, Cham, doi: 10.1007/978-3-030-35983-6_4.

Aragão, B. S., Andrade, R. M. C., Santos, I. S., Castro, R. N. S., Lelli, V., & Darin, T. G. R.. (2021). TestDCat 3.0: catalog of test debt subtypes and management activities. *Software Quality Journal*, doi: 10.1007/s11219-020-09533-y.

Avgeriou, P., Kruchten, P, Nord, R L., Ozkaya, I., & Seaman, C. (2016). Reducing friction in software development. *IEEE Software*, vol. 33, no. 1, pp. 66-73, Jan.-Feb. 2016, doi: 10.1109/MS.2016.13.

Barbosa, L., Freire, S., Rios, N., Ramač, R., Taušan, N., Pérez, B., Castellanos, C., Correal, D., Pacheco, A., López, G., Mandić, V., Maciel, R. S. P., Mendonça, M., Falessi, D., Izurieta, C., Seaman, C., & Spínola, R. (2022). Organizing the TD management landscape for requirements and requirements documentation debt. In *Proceedings of the 25th Workshop on Requirements Engineering* (WER 22).

Behutiye, W. N., Rodríguez, P., Oivo, M., & Tosun, A. (2017). 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, doi: 10.1016/j.infsof.2016.10.004.

Berenguer, C., Borges, A., Freire, S., Rios, N., Ramač, R., Taušan, N., Pérez, B., Castellanos, C., Correal, D., Pacheco, A., López, G., Mendonça, M., Falessi, D., Seaman, C., Mandić, V., Izurieta, C., & Spínola, R. (2023). Investigating the Relationship between Technical Debt Management and Software Development Issues. *Journal of Software Engineering Research and Development*, 11(1), 3:1 – 3:21, doi: 10.5753/jserd.2023.2581.

Berenguer, C., Borges, A., Freire, S., Rios, N., Tausan, N., Ramac, R., Pérez, B., Castellanos, C., Correal, D., Pacheco, A., López, G., Falessi, D., Seaman, C., Mandic, V., Izurieta, C., & Spínola, R. (2021). Technical Debt is not Only about Code and We Need to be Aware about It. In *Proceedings of the XX Brazilian Symposium on Software Quality* (SBQS'21). ACM, New York, NY, USA, 1–12. doi: 10.1145/3493244.3493285.

Cartaxo, B., Pinto, G., Vieira, E., & Soares, S. (2016). Evidence briefings: towards a medium to transfer knowledge from systematic reviews to practitioners. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* (ESEM '16). Association for Computing Machinery, New York, NY, USA, Article 57, pp. 1–10, doi:10.1145/2961111.2962603.

Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén. 2012. Experimentation in software engineering: an introduction. Springer.

Curtis, B., Sappidi, J., & Szynkarski, A. (2012). Estimating the Principal of an Application's Technical Debt. *IEEE Software*, vol. 29, no. 6, pp. 34-42, Nov.-Dec. 2012, doi: 10.1109/MS.2012.156.

Ernst, N. A., Bellomo, S., Ozkaya, I., Nord, R. L., & Gorton, I. (2015). Measure it? Manage it? Ignore it? software practitioners and technical debt. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering* (ESEC/FSE 2015). Association for Computing Machinery, New York, NY, USA, pp. 50–60, doi: 10.1145/2786805.2786848.

Falessi, D., & Kazman, R. (2021). Worst smells and their worst reasons. In *Proceedings of the IEEE/ACM International Conference on Technical Debt* (TechDebt), pp. 45-54, doi: 10.1109/TechDebt52882.2021.00014.

Freire, S., Rios, N., Gutierrez, B., Torres, D., Mendonça, M., Izurieta, C., Seaman, C., & Spínola, R. (2020a). Surveying software practitioners on technical debt payment practices and reasons for not paying off debt items. In *Proceedings of the Evaluation and Assessment in Software Engineering* (EASE '20). ACM, New York, NY, USA, pp. 210–219, doi: 10.1145/3383219.3383241.

Freire, S., Rios, N., Mendonça, M., Falessi, D., Seaman, C., Izurieta, C., & Spínola, R. (2020b). Actions and impediments for technical debt prevention: results from a global family of industrial surveys. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing* (SAC '20). ACM, New York, NY, USA, pp. 1548–1555, doi:10.1145/3341105.3373912.

Freire, S., Rios, N., Pérez, B., Castellanos, C., Correal, D., Ramač, R., Mandić, V., Taušan, N., López, C., Pacheco, A., Falessi, D., Mendonça, M., Izurieta, C., Seaman, C., & Spínola, R. (2021a). How Experience Impacts Practitioners' Perception of Causes and Effects of Technical Debt. In *Proceedings of the IEEE/ACM 13th International Workshop on Cooperative and Human Aspects of Software Engineering* (CHASE). IEEE, Madrid, Spain, 21-30. DOI:10.1109/CHASE52884.2021.00011.

Freire, S., Rios, N., Pérez, B., Castellanos, C., Correal, D., Ramač, R., Taušan, N., Mandić, V., Pacheco, A., López, G., Mendonça, M., Izurieta, C., Falessi, D., Seaman, C., & Spínola, R. (2021c). Pitfalls and solutions for technical debt management in agile software projects. *IEEE Software*, vol. 38 (6), pp. 42-49, doi: 10.1109/MS.2021.3101990.

Freire, S., Rios, N., Pérez, B., Correal, D., Mendonça, M., Izurieta, C., Seaman, C., & Spínola, R. (2021b). How do technical debt payment practices relate to the effects of the presence of debt items in software projects? In *Proceedings of the 28th IEEE International Conference on Software Analysis, Evolution and Reengineering* (SANER), Honolulu (the United States), doi: 10.1109/SANER50967.2021.00074.

Freire, S., Rios, N., Pérez, B., Castellanos, C., Correal, D., Ramač, R., Mandić, V., Taušan, N., López, G., Pacheco, A., Mendonça, M., Falessi, D., Izurieta, C., Seaman, C., & Spínola, R. (2023). Software practitioners’ point of view on technical debt payment. *Journal of Systems and Software*, vol. 196, doi: 10.1016/j.jss.2022.111554.

Freire, S., Rocha, V., Mendonça, M., Izurieta, C., Seaman, C., Spínola, R. (2024a). Assessing IDEA Diagrams for Supporting Analysis of Capabilities and Issues in Technical Debt Management. In: Kadgien, R., Jedlitschka, A., Janes, A., Lenarduzzi, V., Li, X. (eds) Product-Focused Software Process Improvement. PROFES 2023. *Lecture Notes in Computer Science*, vol 14483. Springer, Cham, doi: 10.1007/978-3-031-49266-2_17.

Freire, S., Pacheco, A., Rios, N., Pérez, B., Castellanos, C., Correal, D., Ramač, R., Mandić, V., Taušan, N., López, G., Mendonça, M., Falessi, D., Izurieta, C., Seaman, C., & Spínola, R. (2024b). A Comprehensive View on TD Prevention Practices and Reasons for not Preventing It. *ACM Transactions on Software Engineering and Methodology*, doi: 10.1145/3674727.

Guo, Y., Spínola, R., & Seaman, C. (2016). Exploring the costs of technical debt management – a case study. *Empirical Software Engineering Journal*, vol. 21, pp. 159–182 (2016), doi: 10.1007/s10664-014-9351-7.

Izurieta, C., Vetrò, A., Zazworka, N., Cai, Y., Seaman, C., & Shull, F. (2012). Organizing the technical debt landscape. In *Proceedings of Third International Workshop on Managing Technical Debt* (MTD), Zurich, pp. 23-26, doi: 10.1109/MTD.2012.6225995.

Kruchten, P., Nord, R. L, & Ozkaya, I. (2012). Technical debt: from metaphor to theory and practice. *IEEE Software*, vol. 29, no. 6, pp. 18-21, Nov.-Dec. 2012, doi: 10.1109/MS.2012.167.

Li, Z., Avgeriou, P., & Liang, P. (2015). A systematic mapping study on technical debt and its management. *Journal of Systems and Software*, vol. 101, pp. 193-220, doi: 10.1016/j.jss.2014.12.027.

Mandić, V., Taušan, N., & Ramač, R. (2020). The prevalence of the technical debt concept in Serbian IT industry: results of a national-wide survey. In *Proceedings of the 3rd International Conference on Technical Debt* (TechDebt). ACM, New York, NY, USA, pp. 77–86, doi: 10.1145/3387906.3388622.

Mandić, V., Taušan, N., Ramač, R., Freire, S., Rios, N., Pérez, B., Castellanos, C., Correal, D., Pacheco, A., Lopez, G., Izurieta, C., Falessi, D., Seaman, C., & Spínola, R. (2021). Technical and Nontechnical Prioritization Schema for Technical Debt: Voice of TD-Experienced Practitioners. IEEE Software, vol. 38, no. 6, pp. 50-58, Nov.-Dec. 2021, doi: 10.1109/MS.2021.3103121.

Martini, A. (2018). Anacondebt: a tool to assess and track technical debt. In *Proceedings of the 2018 International Conference on Technical Debt* (TechDebt '18). Association for Computing Machinery, New York, NY, USA, pp. 55–56, doi: 10.1145/3194164.3194185.

Martini, A., Besker, T., & Bosch, J. (2018). Technical debt tracking: current state of practice: A survey and multiple case study in 15 large organizations. *Science of Computer Programming*, vol. 163, pp. 42-61, doi: 10.1016/j.scico.2018.03.007.

McConnell, S. (2007). Technical debt, 10x Software Development Blog. Construx Conversations. URL= http://blogs.construx. com/blogs/stevemcc/archive/2007/11/01/technical-debt-2.aspx.

McHugh, L. M. (2012). Interrater reliability: the kappa statistic. Biochemia medica: Biochemia medica 22(3):276–282.

Mendes, T., Gomes, F., Gonçalves, D. P., Mendonça, M., Novais, R., & Spínola, R. (2019). VisminerTD: a tool for automatic identification and interactive monitoring of the evolution of technical debt items. *Journal of the Brazilian Computer Society*, vol. 25, 2 (2019), doi: 10.1186/s13173-018-0083-1.

Oliveira, F., Goldman, A., & Santos, V. (2015). Managing technical debt in software projects using scrum: an action research. In *Proceeding of 2015 Agile Conference*, National Harbor, MD, USA, 2015, pp. 50-59, doi: 10.1109/Agile.2015.7.

Pacheco, A., Marín-Raventós, G., & López, G. (2019). Technical debt in Costa Rica: an InsighTD survey replication. In: Franch X., Männistö T., Martínez-Fernández S. (eds) Product-Focused Software Process Improvement. PROFES 2019. *Lecture Notes in*

Computer Science, vol. 11915. Springer, Cham, doi: 10.1007/978-3-030-35333-9_17.

Pérez, B., Brito, J.P., Astudillo, H., Correal, D., Rios, N., Spínola, R., Mendonça, M., & Seaman, C. (2019). Familiarity, causes and reactions of software practitioners to the presence of technical debt: a replicated study in the Chilean software industry. In *Proceedings of 38th International Conference of the Chilean Computer Science Society* (SCCC), pp. 1-7, doi:10.1109/SCCC49216.2019.8966424.

Pérez, B., Castellanos, C., Correal, D., Rios, N., Freire, S., Spínola, R., & Seaman, C. (2020). What are the practices used by software practitioners on technical debt payment? results from an international family of surveys. In *Proceedings of the 3rd International Conference on Technical Debt* (TechDebt '20). Association for Computing Machinery, New York, NY, USA, pp. 103–112, doi:10.1145/3387906.3388632.

Pérez, B., Castellanos, C., Correal, D., Rios, N., Freire, S., Spínola, R., Seaman, C., & Izurieta, C. (2021). Technical debt payment and prevention through the lenses of software architects. *Information and Software Technology*, 140, 106692, doi:10.1016/j.infsof.2021.106692.

Ramač, R., Mandić, V., Taušan, N., Rios, N., Freire, S., Pérez, B., Castellanos, C., Correal, D., Pacheco, A., Lopez, G., Izurieta, C., Seaman, C., & Spinola, R. (2022a). Prevalence, common causes and effects of technical debt: Results from a family of surveys with the IT industry. *Journal of Systems and Software*, vol. 184, doi: 10.1016/j.jss.2021.111114.

Ramač, R., Mandić, V., Taušan, N., Rios, N., Mendonça, M., Seaman, C., & Spínola, R. (2020). Common causes and effects of technical debt in Serbian IT: InsighTD survey replication. In *Proceedings of the Euromicro Conference on Software Engineering and Advanced Applications* (SEAA), Portoroz, Slovenia, 2020, pp. 354-361, doi: 10.1109/SEAA51224.2020.00065.

Ramač, R., Taušan, N., Freire, S., Rios, N., Mendonça, M., Spínola, R., & Mandić, V. (2022b). Technical Debt Payment Practices and Rationales Behind Payment Avoidance in the Serbian IT Industry. In: Lalic, B., Gracanin, D., Tasic, N., Simeunović, N. (eds) Proceedings on 18th International Conference on Industrial Systems – IS'20. *Lecture Notes on Multidisciplinary Industrial Engineering*. Springer, Cham, doi: 10.1007/978-3-030-97947-8_14.

Rios, N., Freire, S., Pérez, B., Castellanos, C., Correal, D., Mendonça, M., Falessi, D., Izurieta, C., Seaman, C., & Spínola, R. (2021). On the relationship between technical debt management and process models. *IEEE Software*, doi: 10.1109/MS.2021.3058652.

Rios, N., Mendes, L., Cerdeiral, C., Magalhães, A. P. F., Perez, B., Correal, D., Astudillo, H., Seaman, C., Izurieta, C., Santos, G., & Spínola, R. (2020a). Hearing the voice of software practitioners on causes, effects, and practices to deal with documentation debt. In: Madhavji N., Pasquale L., Ferrari A., Gnesi S. (eds) Requirements Engineering: Foundation for Software Quality. REFSQ 2020. *Lecture Notes in Computer Science*, vol 12045. Springer, Cham, doi: 10.1007/978-3-030-44429-7_4.

Rios, N., Mendonça, M., & Spínola, R. (2018b). 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, doi: 10.1016/j.infsof.2018.05.010.

Rios, N., Mendonça, M., Seaman, C., & Spínola, R. (2019b). Causes and effects of the presence of technical debt in agile software projects. In *Proceedings of the Americas Conference on Information Systems* (AMCIS). Cancún, Mexico.

Rios, N., Spínola, R., Mendonça, M., & Seaman, C. (2018a). The most common causes and effects of technical debt: first results from a global family of industrial surveys. In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* (ESEM '18). Association for Computing Machinery, New York, NY, USA, Article 39, pp. 1–10, doi: 10.1145/3239235.3268917.

Rios, N., Spínola, R., Mendonça, M., & Seaman, C. (2019a). Supporting analysis of technical debt causes and effects with cross-company probabilistic cause-effect diagrams. In *Proceedings of the 2nd International Conference on Technical Debt* (TechDebt). IEEE Press, Piscataway, NJ, USA, pp. 3-12, doi: 10.1109/TechDebt.2019.00009.

Rios, N., Spínola, R., Mendonça, M., & Seaman, C. (2020b). 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*, 25, pp. 3216–3287, doi: 10.1007/s10664-020-09832-9.

Rocha, V., Freire, S., Mendonça, M., & Spínola, R. (2022). Evaluating a Conceptual Framework for Supporting Technical Debt Management in Testing Activities – A Feasibility Study. *Proceedings of the 7th Brazilian Symposium on Systematic and Automated Software Testing* (SAST'22). ACM, New York, NY, USA, 69–78, doi:10.1145/3559744.3559753.

Rocha, V., Freire, S., Rios, N., Lima, C., Ribeiro, L., Perez, B., Dias Neto, A., Moura, H., Correal, D., Mendonça, M., & Spinola, R. (2021). A Conceptual Framework to Support the Management of Technical Debt in Software Testing. In *Proceedings of the Americas Conference on Information Systems* (AMCIS).

Seaman, C. (1999). Qualitative methods in empirical studies of software engineering. *IEEE Transactions on Software Engineering*, 25(4):557-572.

Seaman, C., & Guo, Y. (2011). Measuring and monitoring technical debt. *Advances in Computers*, vol. 82, pp. 25-46, doi: 10.1016/B978-0-12-385512-1.00002-5.

Seaman, C., Guo, Y, Zazworka, N., Shull, F., Izurieta, C., Cai, Y., & Vetrò, A. (2012). Using technical debt data in decision making: potential decision approaches. In *the Third International Workshop on Managing Technical Debt* (MTD), Zurich, Switzerland, 2012, pp. 45-48, doi: 10.1109/MTD.2012.6225999.

Souza, L., Freire, S., Rocha, V., Rios, N., Spínola, R., & Mendonça, M. (2020). Using surveys to build-up empirical evidence on test-related technical debt. In *Proceedings of the 34th Brazilian Symposium on Software Engineering* (SBES '20). ACM, New York, NY, USA, pp. 750–759, doi: 10.1145/3422392.3422430.

Strauss, A., & Corbin, J. M. (1998). Basics of qualitative research: techniques and procedures for developing grounded theory. Sage Publications.

Wohlin, C., Runeson, P., Host, M., Ohlsson, M.C., Regnell, B. & Wesslen, A. (2012). Experimentation in software engineering: An introduction. *Springer*.

Yli-Huumo, J., Maglyas, A., & Smolander. K. (2016). How do software development teams manage technical debt? - an empirical study. In *Journal of Systems and Software*, vol. 120, C (October 2016), pp.195–218, doi: 10.1016/j.jss.2016.05.018.

Appendix

Table 15. TD monitoring-related practices and their examples of citation.

TD Monitoring-related Practice	Quotes from Participants
Adoption of agile methodology	- “Yes, using agile methodologies and tools for that.” - “An agile project methodology was adopted to close gaps, monitor project progress and results.”
Architecture reviewing	- “An analysis of the initial architecture was carried out and an architecture was designed that fulfilled the same functions but was better organized.”
Assign team for TD monitoring	- “In the following sprints, the quality team started to have more monitoring.” - “By the documentation team.”
Code review	- “Every pull request changing the login page must be looked at by least 2 architects.” - “(…) code reviews done during sprint.”
Communicating the stakeholders of TD items	- “The client is perfectly aware of it.” - “It was monitored in the sense that the team had a shared awareness and discussed its ongoing impact.”
Continuous deployment	- “(…) the tools for automated delivery were defined.”
Continuous integration	- “Test with continuous integration, monitoring with sonar etc..”
Cost/benefit analysis	- “We look at the cost of switching and the features available.”
Dashboard	- “Activity tracking.” - “Logs, dashboard and alarms.”
Documentation review	- “Through tools such as code coverage and documentation auditing.” - “In the sprint review, the documentation is checked.”
Focusing on TD payment	- “We were aware of what we are doing and planned to "pay the debt" later on in the maintenance stage of the project.”
Identify the worst debt areas	- “Most of us who are involved with creating and supporting the product are aware of the worst debt areas.”
Identifying TD items	- “We identified the issue we were placed into and when we can address it.”
Improve documentation	- “Forcing proper definition and documentation.”
Improving configuration management practices	- “Code refactored, database versioned, git history cleaner by commit rules.”
Improving planning	- “Better planning and organization.”
Improving software development process	- “Started with a training process, application of agility and the adoption of the best development practices.” - “Through the different reprocesses that occurred.”
Improving tests	- “A test plan was made in conjunction with the user, and they were run again.” - “(…) we began to incorporate more end-to-end tests to gate delivery.”

Improving the requirement management	- "Active management of the requirements to get to implementation quickly."
Infrastructure monitoring	- "The infrastructure sector monitors on-premise servers in the company's custody."
Knowledge sharing	- "With the adoption of Scrum and the use of meetings every day to align the understanding of the development team about the solution that was being developed, and also the practice of pair programming that made it possible to disseminate knowledge."
Measuring the effort	- "We often created timeboxes for how much time we want to spend on updating or maintaining a component. Also, we monitor how much support time we are spending on the app."
Prioritization of TD items	- "To have the user story and prioritize it."
Process automation	- "By automating the process which brought problems to our attention immediately."
Pull request monitoring	- "Through bitbucket functionalities which monitored the pull requests of the user with debt."
Qualified professionals	- "Agile methodologies and Devops were implemented, the team grew, and qualified personnel and specialists were brought in each of their areas."
Quality validation (meetings)	- "Continuously evaluate the quality of the system."
Refactoring	- "Refactoring took place before the development of other services took place." - "Refactoring when appropriate and time allowed."
Risk analysis	- "Before each project, risks were reported to the IT manager."
Support from project management	- "Project management committed to supporting both the user and the developer to minimize the impact on other project tasks."
TD as a task	- "Identified as a task and placed in the backlog for later prioritization." - "Technical debt sub-tasks associated with each task were created."
TD estimation	- "There was a constant estimation of the scope and possible cost of refactoring."
TD item backlog	- "Keep a list of all TD and regularly determine how to eliminate it." - "Identified as a task and placed in the backlog for later prioritization."
TD management plan	- "A schedule was developed specifically to address the case." - "We were aware of what we are doing and planned to 'pay the debt' later on in maintenance stage of the project."
TD status progress report	- "Project manager asked for daily reports on fixes and why things were needing refixed." - "Through daily reports on the progress of projects and possible future stoppers."
Team meetings	- "Regular meeting, usually weekly held that would track the progress." - "(…) the use of meetings every day to align the understanding of the development team about the solution that was being developed (…)."
Team restructuring	- "The work team was structured more efficiently."
Test automation	- "Implementation of automatic tests with requirements for increasing or maintaining coverage, separation of services, integration and continuous deployment, and gradual migration of services and clients."
Tracking TD items	- "Work considered part of the technical debt was tracked." - "Through the task/issue tracking system."
Tracking the cost	- "We tracked the cost of corrective defective data."

Training	- “Beginning with a process of training, application of agility, and appropriation of best practices in development.”
Understanding the cause of TD item	- “Studying the root cause of the problem.”
Use of measuring reports	- “With measurement bulletins.”
Use of metrics for TD identification	- “By introducing code/test metrics.” - “Monitoring based on metrics during the development of the project.”
Use of tools	- “We have to use tool to monitor it, this is an ongoing process.” - “Use of tools like SonarQube.”
Using informal practices	- “Informally. Having high hopes to rectify it by the end of the project.”

Table 16. Practice avoidance reasons and their examples of citation.

Practice Avoidance Reason	Quotes from participants
Business pressure	- "There was no time for that, because of business pressure."
Changes in management	- "Due to repeated changes in leadership."
Changing in the requirements	- "Volatile specifications."
Company with projects beyond capacity	- "More customers than the company's capacity."
Complexity of TD items	- "The technical debt item was too large to monitor and the team had no resources to monitor it."
Cost	- "Because the managers understand that there would be no financial gain, without seeing the maintenance costs."
Effort	- "Because the effort in updating the documentation was very great."
Emotional issues of the team	- "Emotional issues are not something that is generally acknowledged in the programming world let alone in business. You can't bring those types of things up even privately with the leads, you just have to hold it in."
Focusing on short term goals	- "Not in the priority pipeline." - "It was not critical for the success of the project."
Inaccurate time estimate	- "Because the project times were extremely short and the whole team was already doing a lot of overtime."
Ineffective planning and management	- "Ineffective management." - "Lack of management."
Lack of confidence in the technical leader	- "Lack of confidence of the technical manager of the team."
Lack of effort to know the cause of TD	- "In general, they are found guilty, but it is rarely sought to understand the motivational factor of the occurrence."
Lack of experience	- "The team lacks the experience in the new scheme selected to solve the problem and all the strategies result in the same problem."
Lack of interest	- "Little interest of the company to correct this type of situation." - "Management did not care."
Lack of knowledge on TD	- "Because the concept of technical debt was not yet applied in the company." - "There is no knowledge about TD."
Lack of organizational culture	- "This is very dependent on the organization. In this case, it was not identified, monitored, or managed." - "Because there is no permanent initiative to generate changes in the organizational culture."
Lack of qualified professionals	- "Lack of people who could deal with the problem."
Lack of resources	- "Even knowing the problem, there are no resources for immediate solution." - "The time and resources of the project were very limited."
Lack of specific team	- "It was known about for years but we didn't have the headcount to refactor." - "Not enough people or time too."
Lack of TD monitoring process	- "There was no process for it." - "We weren't tracking it."
Lack of TI governance	- "Because there is no IT governance in place, and no PMO in place."
Lack of time	- "Because deadlines are tight." - "The project timeline didn't allow it."
Lack of understanding about the impact of the debt	- "Lack of knowledge of the impact of the TD item in question." - "Technical debt was not identified as a problem at the time."
Late TD identification	- "Too late identification."

Legacy system	- "This project is large and has been developed over years (and is still being actively developed)."
Product delivered	- "After the project is finished, refactorings will not be allowed."
Project delayed	- "(…) all the projects were late."
Project discontinued	- "Because the project simply lost its potential value and gradually became ignored." - "The user team discarded the project."
React when becoming a problem	- "It is neglected until it becomes a problem." - "In the absence of planning, the methodology was reactionary to the problems."
TD Item eliminated as soon as identified	- "We decided to resolve it soon." - "It was not monitored, it was implemented."
TD item payment do not generate revenue	- "Because resolving tech debt is not a revenue-generating. Until it becomes a big enough problem to do something about."
TD was not documented	- "We know that it exists, but not documented."
Team overload	- "There was no time for experts to perform peer reviews because of the amount of work."
Too many TD items	- "Debt occurs with some frequency."

Table 17. Comparison to related work on TD monitoring-related practices.

	Our Study	**Technical Debt Monitoring-Related Practices from**										
		RW1	**RW2**	**RW3**	**RW4**	**RW5**	**RW6**	**RW7**	**RW8**	**RW9**	**RW10**	**RW11**
Enabling TD monitoring	Adoption of agile methodology	-	-	-	-	Implementing pair programming or test-driven development	-	-	-	-	-	-
	Assign team for TD monitoring	-	-	Defining a responsible for monitoring each identified and measured TD item	-	-	-	-	-	-	-	-
	Continuous integration	-	-	-	-	Continuous integration tools	-	-	-	-	-	-
	Improving planning	-	-	-	-	-	-	-	-	-	-	-
	Improving software development process	-	-	-	-	-	-	-	-	-	-	-
	Improving the requirement management	-	-	-	-	-	-	-	RE-KOMBINE model	-	-	-
	Infrastructure Monitoring	-	-	-	-	-	-	-	-	-	-	-
	Knowledge sharing	-	-	-	-	-	-	-	-	-	-	-
	Process automation	-	-	-	-	-	-	-	-	-	-	-
	Pull request monitoring	-	-	-	-	-	-	-	-	-	-	-
	Quality validation (meetings)	-	-	-	-	-	-	-	-	-	-	-
	Support from project management	-	-	-	-	-	-	-	-	-	-	-
	TD management plan	Accounting	-	-	-	Planning in advance for TD	-	-	Accounting formal approach to	-	-	-

									TD decision making			
	Team restructuring	-	-	-	-	-	-	-	-	-	-	-
	Use of tools	-	-	-	Using data collected from (management or TD measuring) tools	-	AnaConDebt	Using system for bug fixing	Measuring symptom severity on a smell thermometer Sonar TD plugin DebtFlag TD evaluation (SQALE) Software maps tool Code Christmas tree	VisminerTD	tools	-
Monitoring action	Communicating the stakeholders of TD items	-	-	-	-	-	-	-	-	-	-	-
	Cost/benefit analysis	Cost/benefit	-	-	-	-	-	-	Cost-benefit analysis	-	-	Monitor changes in the cost/benefit ration of the identified debt
	Dashboard	Portfolio management	-	-	-	Collective dashboards	-	-	Portfolio approach	-	-	-
	Identify the worst debt areas	-	-	-	-	-	-	-	Debt symptoms index	-	-	-
	Measuring the effort	Real options marketing	-	-	-	-	-	-	Options marketing	-	-	-
	Prioritization of TD items	-	-	-	-	-	-	-	-	-	-	-
	Risk analysis	-	TD monitoring as part of risk process	-	-	-	-	-	-	-	-	-
	TD as a task	-	-	-	-	-	-	-	-	-	-	-

	TD estimation	-	-	-	-	Improving estimation techniques of sprints	-	-	-	-	-	-
	TD item backlog	-	Backlog grooming	-	-	-	Creation of TD items in a backlog	Reporting TD items in backlog	-	Panels (TO DO, DOING, and DONE), based on the Kanban concept	-	-
	TD status progress report	-	-	-	-	Visualization techniques	-	-	-	-	-	-
	Team meetings	-	-	-	-	-	-	-	-	-	-	-
	Tracking TD items	-	-	-	-	-	-	-	-	-	-	-
	Tracking the cost	-	-	-	-	-	-	-	-	-	-	-
	Understanding the Cause of TD Item	-	-	-	-	-	-	-	-	-	-	-
	Use of measuring reports	-	-	-	-	-	-	-	-	-	-	-
	Using informal practices	-	-	-	-	-	-	-	-	-	Manual monitoring	-
TD identification	Identifying TD items	-	-	-	-	-	-	Statically analyzing the code for finding TD items or potential bugs, or security issues	-	-	-	-
	Use of metrics for TD identification	-	-	-	-	-	-	-	Metrics for managing architectural TD	-	-	-
TD	Focusing on TD payment	-	-	-	-	-	-	-	-	-	-	-

repayment	Improve documentation	-	-	-	-	-	-	-	-	-	-	-
	Refactoring	-	-	-	-	-	-	-	-	-	-	-
TD prevention	Architecture reviewing	-	-	-	-	-	-	-	-	-	-	-
	Code review	-	-	-	-	-	-	-	SQALE method	-	-	-
	Continuous deployment	-	-	-	-	-	-	-	-	-	-	-
	Documentation review	-	-	-	-	-	-	-	-	-	-	-
	Improving configuration management practices	-	-	-	-	-	-	-	-	-	-	-
	Improving tests	-	-	-	-	-	-	Measuring test coverage	-	-	-	Changes in the test process
	Qualified professionals	-	-	-	-	-	-	-	-	-	-	-
	Test automation	-	-	-	-	-	-	-	-	-	-	-
	Training	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	setting a commonly agreed definition of done	-	-	-	-	-	-
-	-	-	-	-	-	-	-	using comments in the code or other artifacts	-	-	-	-
-	-	-	-	-	-	-	-	documenting issues in text or spreadsheets	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	monitor triggers
-	-	-	-	-	-	-	-	-	making of dependencies and code problems	-	-	-

-	-	-	-	-	-	-	-	-	supply chain management	-	-	-

Caption:	RW3: Oliveira *et al*. (2015)	RW6: Martini (2018)	RW9: Mendes *et al*. (2019)
RW1: Ampatzoglou *et al*. (2015)	RW4: Yli-Huumo *et al*. (2016)	RW7: Martini *et al*. (2018)	RW10: Apa *et al*. (2020)
RW2: Ernst *et al*. (2015)	RW5: Behutiye *et al*. (2017)	RW8: Rios *et al*. (2018b)	RW11: Aragão *et al*. (2021)