Assessing IDEA Diagrams for Supporting Analysis of Capabilities and Issues in Technical Debt Management

Sávio Freire[1,2][0000-0002-3989-9442], Verusca Rocha[1][0000-0002-1307-985X],
Manoel Mendonça[1][0000-0002-0874-7665], Clemente Izurieta[3][0000-0002-1002-3906],
Carolyn Seaman[4][0000-0001-6588-9830], and Rodrigo Spínola[5][0000-0003-0272-9578]

[1] Federal University of Bahia, Salvador, Bahia, Brazil
[2] Federal Institute of Ceará, Morada Nova, Ceará, Brazil
[3] Montana State University, Bozeman, MT, United States
[4] University of Maryland Baltimore County, Baltimore, MD, United States
[5] Virginia Commonwealth University, Richmond, VA, United States
savio.freire@ifce.edu.br, veruscacrocha@gmail.com,
manoel.mendonca@ufba.br, clemente.izurieta@montana.edu,
cseaman@umbc.edu, spinolaro@vcu.edu

Abstract. *Context*. Technical debt management (TDM) comprises activities such as prevention, monitoring, and repayment. Current technical literature has identified, for each of these TDM activities, several applicable practices as well as practice avoidance reasons (PARs). This body of knowledge (practices and PARs) is available in the literature only in widely spread text and tables, and is not organized into artifacts, hindering the use of current knowledge on TDM. Previously, we organized these practices and PARs into IDEA (Impediments, Decision factors, Enabling practices, and Actions) diagrams. However, an empirical evaluation of these diagrams is still missing. *Aims*. To empirically assess the IDEA diagrams with respect to their ease of use, usefulness, potential future use, and support for TDM activities. *Method*. We conduct two complementary empirical studies. Firstly, we applied the technology acceptance model (TAM) with 72 participants in academic contexts. Afterwards, we interviewed 11 experienced software practitioners. *Results*. In the TAM study, 92% of the participants indicated that they could use the diagrams. Also, the diagrams were considered easy to learn and use. Through the interviews, participants indicated that the diagrams are easy to read and follow, can influence decisions on how to manage debt items, and could be used to support their daily activities. *Conclusion*. Both studies provided positive evidence that IDEA diagrams can be useful for supporting TDM activities.

Keywords: Technical Debt, Technical Debt Management, IDEA Diagrams.

1 Introduction

Technical debt (TD) emerges from intentional shortcuts or even mistakes taken by software practitioners in their projects [1, 2]. Incurring debt can bring short-term benefits, usually in terms of high productivity, but also long-term drawbacks, making the software difficult to evolve [3, 4, 5]. By performing TD management activities, a software team can make TD items visible and under control, allowing it to balance benefits and drawbacks of debt presence [6].

TD management comprises several activities, such as prevention, monitoring, and repayment [6, 7]. By performing TD prevention, software teams can avoid potential TD items, while TD monitoring follows the identified TD items to measure their cost/benefits along with their elimination. This elimination is performed during the repayment activity. Knowing the practices to prevent, monitor, and repay debt items can support software teams in choosing the most appropriate practices in their context. On the other hand, and for different reasons, teams sometimes avoid the application of these practices. Having information of these reasons (herein called *practice avoidance reasons* - PARs) can aid software teams in increasing their ability in TD management, revealing internal or external factors resulting in TD non-prevention, non-monitoring, and non-repayment.

Related work has investigated TD prevention, monitoring, and repayment practices and PARs [6, 8-17]. For instance, Bomfim Jr and Santos [9] identified TD repayment practices and PARs considered in the agile software development process. Rios et al. [15] identified prevention and repayment practices for managing documentation debt items, while Aragão et al. [16] investigated prevention, monitoring, and repayment practices for test debt items. Despite the valuable contributions of the current literature in the area, there is still a need to organize the current body of practices and PARs into artifacts that can effectively be applied to support the management of TD in software projects. Such an artifact could provide guidance on how to understand and select the practices or PARs in isolation as well as in combination. In the absence of this guidance, development teams rely on textual information spread through several tables, thus hindering the use of current knowledge on TD management.

We propose to help fill this gap by using IDEA (Impediments, Decision factors, Enabling practices, and Actions) Diagrams to organize information on TD prevention, monitoring and repayment practices and PARs [17]. Loosely inspired by SWOT (strengths, weaknesses, opportunities, and threats) analysis [18], the IDEA diagrams organize TD management practices and PARs into quadrants: capabilities (actions and enabling practices) and issues (decision factors and impediments). To populate them, we use the practices and PARs reported by 653 practitioners who responded to the InsighTD survey (http://td-survey.com), which is a globally distributed family of industrial surveys on TD [19].

In this work, we go further and investigate to what extent the IDEA diagrams can support software teams in increasing their capability for preventing, monitoring, and repaying the debt. We investigate if the diagrams are useful, their ease of use, whether they can influence decisions about how to manage debt items, whether they can be used in daily project activities, and their potential future use. We empirically investigate IDEA diagrams through two complementary studies. Initially, we applied the Technology Acceptance Model (TAM) [20] with 72 students enrolled in a software engineering course. Results indicate that the diagrams can positively support TD management, making it easier to identify practices and PARs associated with TD prevention, monitoring, and repayment activities. Also, 92% of the participants stated that they could use the IDEA diagrams to manage TD items. Subsequently, we conducted an interview-based case study with 11 software practitioners. The participants pointed out that the IDEA diagrams are easy to read and follow, can influence the decisions on TD management, and could provide useful guidance if used. The results from both studies indicate that the IDEA diagrams are sound and can be used to increase the capability of software teams to manage debt items.

Beyond this introduction, this paper has six more sections. Section 2 discusses related work on TD prevention, monitoring, and repayment, and the IDEA diagrams. Next, sections 3 and 4 present the TAM and interview study we performed to assess the diagrams, respectively. We discuss the results in Section 5. Section 6 discusses the threats to validity. Finally, Section 7 presents the final remarks and future work.

2 Background

In this section, we initially discuss related work on TD prevention, monitoring, and repayment. We then present the IDEA diagrams.

2.1 Related Work on TD Prevention, Monitoring and Repayment

Technical literature reveals investigations into the prevention, monitoring, and repayment of debt items. By performing a systematic literature review, Li et al. [6] identified a set of categories for TD prevention, TD monitoring, and TD repayment, while Behutiye et al. [10] recognized a set of monitoring practices and categories for TD payment in agile software development processes.

By conducting case studies in industrial settings, Yli-Huumo et al. [8] identified some TD prevention practices (e.g., *coding standards*) and one practice (*used data collected from (management or TD measuring) tools*) for monitoring the debt. Bomfim Jr and Santos [9] identified a set of TD prevention (e.g., *using coding standards*), monitoring (e.g., *including TD tasks in product backlog*), and repayment (e.g., *refactoring older code*) practices in agile software development processes. The authors also identified the reasons (e.g., *low impact for business* and *high effort*) which hamper the application of those practices. Toledo et al. [11] identified 13 repayment practices used to eliminate architecture debt in microservices, such as *rewrite the communication layer*.

Silva et al. [12] ran a survey and identified TD prevention (e.g., *retrospective meetings*) and repayment (e.g., *redesign*) practices. Two replications of this survey were performed [13, 14], finding the same preventive practices previously reported and confirming the repayment practices. Rios et al. [15] recognized preventive (e.g., *comment the code*) and repayment (e.g., *review outdated documentation*) practices for documentation debt items. Lastly, Aragão et al. [16] identified a set of TD prevention (e.g., *present already identified debts*), monitoring (e.g., *changes in the test process*), and repayment (e.g., *change test cases by analyzing defects*) practices for test debt items.

Although these studies revealed practices used to prevent, monitor, or repay TD, most of them did not provide an artifact that organizes these practices and supports software practitioners in effectively managing TD items. Without an artifact, software practitioners rely on textual information spread through several tables. To deal with this gap, we proposed the IDEA diagrams, which are presented in next subsection.

2.2 The IDEA Diagrams

IDEA diagrams are inspired by SWOT (strengths, weaknesses, opportunities, and threats) analysis [18] to organize issues (decision factors and impediments) and capabilities (actions and enabling practices) related to TD management into four quadrants. Unlike SWOT, the scope of the IDEA diagrams is not organizational planning but is to support software teams in increasing their ability to manage the debt [17]. The diagrams can be defined for any TD management activity and their practices and PARs can be specialized considering the types of debt (such as code, design, and requirements) and project context variables, such as process model. We presented a set of IDEA diagrams for agile software development processes in [17].

Fig. 1-A presents the diagram's structure and how the quadrants are related to each other. Each quadrant is depicted by a specific color and contains a set of practices or PARs. On the left side of the diagram, practices are concentrated in the actions and enabling practices quadrants. Actions (in the upper left quadrant) are practices or techniques that, when employed, will have a direct effect on TD management. Enabling practices (lower left), on the other hand, have an indirect effect on a team's ability to effectively manage TD by enabling a culture that promotes TD management or providing resources that are important for effective TD management. On the right side, the diagram presents the PARs in the decision factors and impediments quadrants. The decision facts (in the upper right quadrant) represent factors that led to decisions explicitly made by the team itself to incur TD or to not pay off TD. Impediments (lower right) are conditions or decision originating from an external agent (i.e., a customer or organization) that are outside the control of the project team, but that make it difficult or impossible to manage TD effectively. In all quadrants, the practices and PARs are ordered by a criterion that can be defined by software teams. For example, a sorting criterion could be how frequently practices and PARs have been used in the project in the past.

We used data from the InsighTD project to define IDEA diagrams for TD prevention, monitoring, and repayment. Also, we specialized them for design and documentation debt. Fig. 1-B shows the IDEA diagram for design debt repayment with the five most cited elements per quadrant. The complete version is available at https://bit.ly/3NvrzPw. The percentages with practices and PARs inform how frequently they were used in the InsighTD participants' software projects.

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

Analyzing the relationships between quadrants can support software teams in boosting their TD management initiatives. Looking at (Fig. 1-B):

- **Actions and Enabling practices quadrants** can provide teams with a way to increase their TD management capacity by suggesting enabling practices that could support actions they already employ. For example, suppose a software team already uses *code refactoring* and *design refactoring* actions to repay design debt items, but not as often as they could. By discussing potential enabling practices, they could realize that *investing effort on TD repayment activities* and *negotiating deadline extension* would enable them to employ their successful refactoring tasks more often.
- **Decision factors and Impediments quadrants** can support teams to understand why they are not managing TD. For example, a software team might identify that they often decide against repaying TD because they are *focusing on short term goals.* By digging a bit deeper and discussing items in the impediments quadrant, they might realize that the *team overload* impediment is the primary reason for the short-term focus, and thus for not repaying design debt items. This could help equip them to take steps with their customers to argue for easing the load on the team.
- **Enabling practices and Decision factors quadrants** can reduce weak areas related to TD management. Suppose a team realizes that the *lack of adoption of lessons learned* decision factor is often the reason for design debt non-repayment. Then, the team could examine enabling practices that could counter their tendency to fail to apply lessons learned in TD decision-making, such as *improving software development process* and *improving the team collaboration*.
- **Actions and Impediments quadrants** can help teams to reduce the impediments for TD management. For example, if a team identifies that *complexity of the project* impediment hampers the payment of design debt, the team can apply *code refactoring* or *design refactoring* actions for reducing the complexity of the project, reducing the external factors in TD repayment decisions.

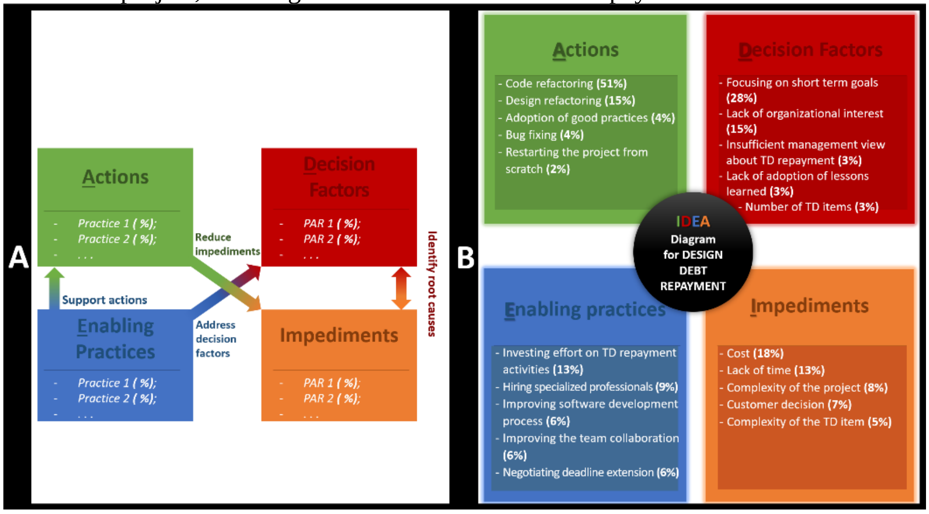
A
Actions
- Practice 1 (%);
- Practice 2 (%);
- . . .
Decision Factors
- PAR 1 (%);
- PAR 2 (%);
- . . .
Reduce impediments
Identify root causes
Support actions
Enabling Practices
- Practice 1 (%);
- Practice 2 (%);
- . . .
Address decision factors
Impediments
- PAR 1 (%);
- PAR 2 (%);
- . . .
B
Actions
- Code refactoring (51%)
- Design refactoring (15%)
- Adoption of good practices (4%)
- Bug fixing (4%)
- Restarting the project from scratch (2%)
Decision Factors
- Focusing on short term goals (28%)
- Lack of organizational interest (15%)
- Insufficient management view about TD repayment (3%)
- Lack of adoption of lessons learned (3%)
- Number of TD items (3%)
IDEA Diagram for DESIGN DEBT REPAYMENT
Enabling practices
- Investing effort on TD repayment activities (13%)
- Hiring specialized professionals (9%)
- Improving software development process (6%)
- Improving the team collaboration (6%)
- Negotiating deadline extension (6%)
Impediments
- Cost (18%)
- Lack of time (13%)
- Complexity of the project (8%)
- Customer decision (7%)
- Complexity of the TD item (5%)

Fig. 1. (A) The IDEA diagram's structure. (B) A summarized version of the IDEA diagram for design debt repayment.

For assessing the IDEA diagrams in terms of their support of TD prevention, monitoring, and repayment activities, we performed two empirical studies which are presented in Sections 3 and 4.

3 Assessing the Ease of Use, Usefulness, and Potential Future Use of the IDEA Diagrams

The goal of this study was to **analyze** the IDEA diagrams **for the purpose of** characterizing them **with respect to** ease of use, usefulness, and potential future use **from the point of view of** undergraduate students enrolled in a software engineering course **in the context of** software development projects. As our intention was to investigate the perception on the use of a new technology (IDEA diagrams), we applied the technology acceptance model (TAM) [20]. It captures the opinion of the participants on three constructs (perceived usefulness, ease to use, and self-predicted future use), measured through a set of questions.

3.1 Project Context

The study consisted of analyzing the ease of use, usefulness, and potential future use of the IDEA diagrams through the simulation of TD management activities, whose objective was to identify, from a list of TD items, the prevention, monitoring and repayment practices and PARs that could be applied for the project. The list of debt items was extracted from a real software project called National Transplantation System (NTS), developed by a partner organization (the Fraunhofer Project Center at the Federal University of Bahia).

The NTS is responsible for the control and monitoring of transplants of organs, tissues and parts of the human body for therapeutic purposes in Brazil. The product consists of a medium-large database-driven web application. It includes several modules distributed through 212 use cases. The application was written in Java and based on the MVC framework. It includes 365K lines of code in 1377 domain classes. The project was developed with the following infrastructure: Eclipse IDE, Subversion, and Trac. The development team was composed of 1 project manager, 1 technical leader, 3 requirements analysts, and 8 developers. The project followed a Scrum-like development process to continuously integrate features and deliver working versions to the customer. The project team manually identified TD items and organized them into a spreadsheet, which constituted the list of TD items used in this study.

3.2 Procedure and Instrumentation

Initially, the participants filled in a characterization and consent form. Then, the first author trained the participants on TD and its concepts associated with the case study, such as, TD definition, design and documentation debt, and TD prevention, monitoring, and repayment activities. An example of identification of practices and PARs related to those activities was also explained (**step 1**). As we wanted to reduce bias during the identification of TD management practices and PARs step of the study, we used an example in the context of house maintenance. For instance, a repayment practice for *the kitchen drain is showing a slow flow of water* could be *using a plunger*.

In **step 2**, the participants, individually, analyzed in an *ad hoc* manner a design debt item and a documentation debt item (see Table 1) to suggest practices and PARs associated with the TD prevention, monitoring, and repayment of those items. We chose particular debt items that were described in detail in the list of TD items provided by our industry partner.

Step 3 focused on the training on the IDEA diagrams, explaining how to use them to support the analysis of practices and PARs associated with TD prevention, monitoring, and repayment. We presented an example of how to analyze a debt item using the diagram, but also in the context of the house maintenance scenario.

Table 1. TD items used in the case study.

Step	TD type	TD item description
2	Documentation	The allocation module does not have a req. specification document.
2	Design	A verification with the name of the activity is necessary when it is required to identify a type of service or bill. This information is fixed in the code and can bring errors when some update is performed, or the data in the database has incorrect names.
4	Documentation	The documentation should be up-to-date and requirements gathering should be conducted in accordance with the customer's needs. Frequent changes to these modules caused a lot of rework.
4	Design	The invoice printing functionality need to be simplified. The functionality is working correctly but needs to be adjusted in the future to be more adherent to the system design.

In **step 4**, the participants received two new TD items, shown in Table 1, and analyzed them using the IDEA diagrams to suggest practices and PARs associated with the prevention, monitoring, and repayment of those items. The participants received a set of IDEA diagrams for the types of analyzed debt items (documentation or design) for each TD management activity.

Lastly, the participants individually completed the evaluation form, containing a set of questions associated with the three constructs (perceived usefulness, ease to use, and self-predicted future use) considered in the TAM (**Step 5**). The evaluation form is available at https://bit.ly/3NvrzPw. To answer the questions in the form, the participants indicated the option that best represented their point of view on the IDEA diagrams, according to the following 5-point scale: (1) I totally agree; (2) I agree partially; (3) Neutral; (4) Partially disagree; and (5) Strongly disagree. At the end of the form, the participants also described the positive and negative aspects of the diagrams and suggestions for improvements and indicated whether the diagrams helped them to identify practices and PARs that they would not have identified without using them.

3.3 Data Analysis

All answers were validated by following three criteria: (i) the participant filled in the consent and characterization forms, (ii) the participant performed the two activities of analysis of TD elements (steps 2 and 4), and (iii) the participant filled in the evaluation form.

For the closed questions, we calculated the share of participants choosing each option to obtain a better understanding of the data. For open-ended questions, we applied a coding process to identify the central idea described in the answers [21, 22]. For example, a participant indicated the following positive aspect of the IDEA diagrams: "items properly separated and placed, easy to locate." As this answer is related to the diagram representation, we coded it as *adequate representation structure*. The coding process was performed by the first author and revised by the last author. Divergences were resolved in a consensus meeting. In the end, we had a list of codes and their respective number of occurrences.

3.4 Results

The participants were undergraduate students enrolled in a software engineering course. In total, 72 participants completed all required steps. About 19% of them indicated that have at least one year of experience with software development. Participants also indicated their level of experience in nine areas related to the software development process. We present the results in Table 2. We can notice that there are participants with experience in all areas of software engineering. Lastly, most of the participants had some level of knowledge on TD ranging from low (53%) to good (10%) and expert (3%).

Table 2. Level of Experience of Participants.

Knowledge area	Level of experience*				
	1	2	3	4	5
Project management	23	33	8	2	6
Monitoring and correction of software defects	29	27	4	4	8
Software maintenance	31	22	5	5	9
Software architecture	28	28	5	6	5
Software design	22	28	5	9	8
Software documentation	27	34	2	2	7
Requirement specification	20	40	7	1	4
Implementation	19	16	5	22	10
Software testing	22	28	6	8	8

* Levels of experience: (1) none, (2) studied in class, (3) practiced in classroom projects, (4) used in personal projects, and (5) used in projects in the industry.

Participants' Point of View on IDEA Diagrams for TD Prevention, Monitoring, and Repayment. The TAM statements for the **perceived usefulness** construct asked each participant whether they are able to (U1) identify practices or PARs more quickly, (U2) improve their performance in identifying practices or PARs, (U3) improve their effectiveness in identifying practices or PARs, and (U4) make easy to identify practices or PARs, compared to carrying out these tasks without the IDEA diagram. These constructs are related to each TD management activity. Most of the participants agreed with the affirmations for IDEA diagrams for TD prevention (more than 86% of the participants), TD monitoring (more than 81%), and TD repayment (more than 89%). Thus, comparing the task execution with and without IDEA diagrams, the participants had high productivity (U1), increased performance (U2 and U4), and efficacy (U3). Moreover, 90% of the participants agreed with the following statements: “using the diagrams, I would increase my productivity in identifying practices and PARs” (strongly agree: 65%, agree: 25%, and neutral: 10%) and “I believe the proposed diagrams would be useful to support technical debt management” (strongly agree: 72%, agree: 18%, and neutral: 10%). All detailed results are available at https://bit.ly/3NvrzPw.

Fig. 2 presents TAM statements for the **ease-of-use** construct. At least 80% of the participants agreed with the statements associated with the benefits: easy to learn (E1), clear and understandable (E2), easy to use for particular tasks (E3-E8), easy to become skillful (E9), easy to remember (E10), and easy to use (E11).

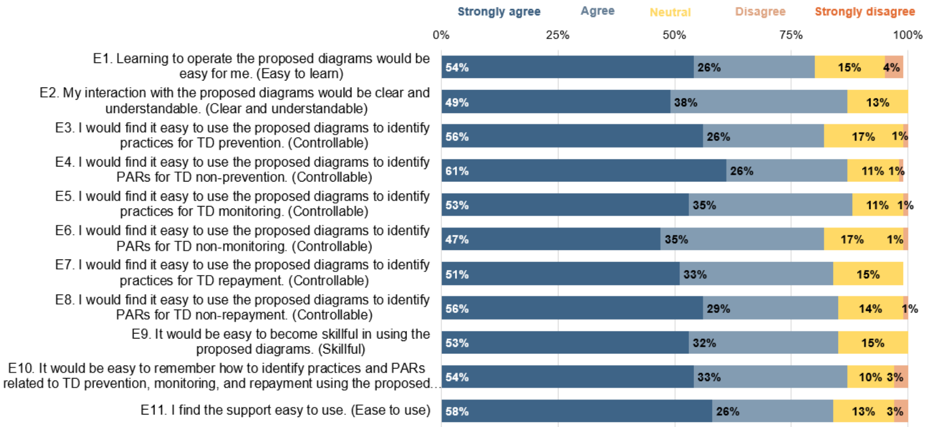

Fig. 2. Ease-of -use items for IDEA Diagrams for TD Prevention, Monitoring, and Repayment.

The participants also provided their opinion for the **self-predicted future use** construct. We found that (i) 92% of the participants agreed with “Assuming the proposed diagrams would be available to manage technical debt, I would use them in the future” (strongly agree: 63% and agree: 29%) and (ii) 63% of the participants agreed with “I would prefer to use the proposed diagrams to identify practices and PARs associated with TD prevention, monitoring and repayment activities than in the usual way (without the diagrams). Only 15% of the participants disagreed with this statement (strongly agree: 45%, agree: 18%, neutral: 22%, disagree: 7%, and strongly disagree: 8%).

Lastly, most of the participants indicated that using the diagrams helped them identify (i) TD prevention, monitoring, or repayment practices (93% of the participants) and (ii) PARs for TD non-prevention, non-monitoring, and non-repayment (94% of the participants) that they would not have identified without the diagrams.

Positive Points Reported. The participants indicated that the IDEA diagrams allow an *easy identification of practices and PARs* (number of occurrences (NO) = 34), as described in the participant's quote "it is easier to identify reasons and practices for prevention, monitoring, and repayment…" The participants also explained that the IDEA diagrams have an *adequate representation structure* (NO = 25), for instance, "detailed, items properly organized, easy to locate." Lastly, participants argued that the diagrams *facilitate the decision making* (NO = 6; e.g., "better performance and effectiveness in decisions."). Other mentioned positive points are *ease of use, provide a variety of practices and PARs,* and *facilitate the understanding*.

Negative Points Reported. Participants indicated that *it is possible that practitioners only consider the practices/PARs in the diagram* (NO = 10), as illustrated in "it can create a false impression that everything has been listed and cause a lack of interest in identifying other elements". Also, the participants affirmed that *the diagrams have many practices or PARs* (NO = 5; e.g., "it contains a lot of information..."). Other negative points mentioned were *they do not present all practices and PARs* and *lack of dynamic manipulation of the diagram*.

Improvement Points Reported. The participants suggested the following improvements: (i) *better organize information* (NO = 10; "better distribution of data in the space in each quadrant."), (ii) *enable dynamic manipulation of the diagram* (NO = 4; "there could be some way to navigate through the diagrams..."), (iii) *enable the inclusion of new elements* (NO = 4; "Open a checkbox with the option to include new reasons."), (iv) *simplify the diagram* (NO = 3; "use diagrams as simple as possible."), (v) *remove redundant practices and PARs* (NO = 1; "some items can be merged as long as they look similar."), and (vi) *better explain percentages* (NO = 1; "explain the percentages present in the diagrams.").

4 Perception of Software Practitioners

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

4.1 Procedure and Instrumentation

We conducted semi-structured individual interviews composed of three steps. In the first step (**opening**), we presented the consent form and the concept of TD. Then, the participant answered questions on their expertise in TD management, such as the level of experience with TD management and the strategies and tools they have used to manage debt. In the second step (**perception about the IDEA diagrams**), we presented the IDEA diagrams and provided some examples of using the diagrams for supporting TD management. Then, we asked participants whether the diagrams (i) would be easy to use and follow, (ii) could influence their decision about how to manage the debt, and

(iii) could be used in their daily activities. In the last step (**closing**), we asked participants if they have anything more to say about the diagrams and asked them to fill in a characterization form. The interview script and the participant characterization form are available at https://bit.ly/3NvrzPw.

The first author conducted all interviews remotely. Each of them lasted around 30 minutes and were recorded with the interviewee's permission.

4.2 Data Analysis

We transcribed the interviews and organized the answers by question. Then, we coded the transcripts to identify the main idea presented in each answer [21, 22]. For example, a participant explained why IDEA diagrams can be used in daily activities: "you can communicate better with your team to avoid future problems." We coded it as *assist team communication*. This process was performed by the first author and revised by the last author. Divergences were resolved in a consensus meeting. Finally, we had a list of codes and their respective number of occurrences.

Concerning the characterization questions, we calculated the share of participants choosing each option of the characterization form to obtain a better understanding of the data. The characterization form is available at https://bit.ly/3NvrzPw.

4.3 Results

We invited 11 practitioners from our contacts in the software industry. Most of them work in medium-sized companies (organizations with 51 to 1000 employees; 6 participants), followed by large (more than 1000 employees; 4 participants) and small (up to 50 employees; 1 participant). The participants identified themselves as project manager or leader (3 participants), developer (2), product owner (2), process analyst (1), agile coach (1), tester (1), and software architect (1). Regarding the participants' experience level, we interviewed 5 experts (authoritative knowledge of discipline and deep tacit understanding across area of practice), 5 proficient (depth of understanding of discipline and area of practice), and 1 competent (good working and background knowledge of area of practice). The participants mostly adopted agile software development (7) and the others followed hybrid methodologies.

Three participants did not have previous experience on TD management, while eight of them have experience by participating in projects in which they have identified TD items or have tried to actively manage them. The identified debt items are commonly registered in the product backlog or in managerial tools.

Easy to read and follow. Most of the participants (nine participants) affirmed that the IDEA diagrams are easy to read and follow to support decisions on TD management because the diagrams: (i) *facilitate TD decision making* ("Because you can extract data from their topics (actions, impediments, decisions...). They are very visible in aiding decision making"), (ii) *are succinct and clear* ("Yes, because I think they are very succinct and clear"), (iii) *can be understood by all stakeholders* ("I also clearly see how to use them in a very didactic way, even the product owner could understand"), (iv) *present in a summarized way both internal and external issues* ("I can see how I can get an x-ray of internal and external issues that are still not leading me to manage well debt items"), (v) *can be used in reviewing and planning meetings* ("...artifact to be considered at each review and backlog planning meeting"), and (vi) *facilitate TD items identification* ("It makes it easier to perceive the TD"). Lastly, three of these participants warned that the diagrams are *easy to use but are not self-explanatory* ("…Having the explanation is very useful to have visibility and put them into practice").

Influence decision about how to manage the debt. Only one participant indicated that the diagram would not influence his/her decisions ("…in my case not so much. I have already implemented some of the practices you mentioned there"). The other participants reported that the diagrams could influence their decisions. They explained that the diagrams (i) *facilitate the communication between stakeholders* ("even with people who are not part of the team, you can take a picture of the situation and try to negotiate strategies to improve it."), (ii) *support the decision making on TD items* ("…from that diagram, make decisions about what would be relevant to do"), (iii) *support to identify problems* ("I would have clarity of the reasons that prevent me from managing them."), (iv) *have a customizable catalogue of practices used in the software industry* ("A catalog of best practices could be customized for each team."), and (v) *allow an effective risk management* ("As if it was an effective risk management, but for debt management. I can map impediments and internal factors and at the same time put together this action plan to improve management").

In addition, almost all participants (nine) indicated that the percentages would be useful for choosing a practice or a PAR, highlighting that they *support the practices and PARs prioritization*, *present the most representative elements*, and *are based on previous experience*. The other two participants mentioned that percentages can be difficult to calculate. Lastly, one participant was unsure about the usefulness of the percentage because it represents the consensus of other organizations, which not necessarily is related to her/his current context. However, the same participant indicated that the diagrams could be adapted to her/his context.

Can be used in daily activities. All participants indicated that they could use the IDEA diagrams to support TD management activities. The participants explained that the diagrams (i) *enable continuous improvement of TD management actions* ("I see very clearly their use within a team, having a complete view of management and allowing us to set up a continuous improvement plan of actions to improve management"), (ii) *assist in tracking TD items*, (iii) *indicate possible problems and solutions to resolve them*, and (iv) *assist team communication*.

Most of the participants (six) indicated that the diagrams could be adapted to their current context because they *would assist in negotiating project constraints* and *highlight the problems*. The participants also indicated the following necessary adjustments in the diagrams: (i) *remove practices that do not fit the developer's scope* ("I have a programmer's point of view. I am not on the manager side. I would cut some things out to make the set of actions more streamlined"), (ii) *include arrows between quadrants to indicate how the analysis should be done*, and (iii) *make it automated by suggesting relationships between quadrants*.

5 Discussion

From the TAM study conducted with undergraduate students, we sought to collect initial evidence on the usefulness of IDEA diagrams. We found that 92% of the participants indicated that they could use the diagrams. Most of the participants also agreed that, by using them, they can see productivity gains in performance, and effectiveness in performing the task proposed in the study. Also, the diagrams were considered easy to learn and use. Based on the positive results, we decided to approach software practitioners. In the interviews, most of the participants indicated that the diagrams would be easy to read and follow, could influence decisions on how to manage debt items, and could be used to support daily activities.

In summary, the TAM study and the interviews provided positive evidence that the IDEA diagrams can be useful for supporting TD management activities. By identifying the actions and enabling practices, practitioners can define strategies for boosting their TD management activities, while having information on decisions factors and impediments can support practitioners in defining strategies for reducing the internal and external factors that result in TD non-management. By analyzing the relationships between quadrants, diagrams can assist practitioners in defining these strategies.

Results also provide initial evidence that the IDEA diagrams can be used by practitioners with or without experience in managing TD items. For software teams who want to start managing TD, the ranked lists of practices and PARs organized in each of the IDEA diagrams can provide guidance on what to employ (practices) or curb (PARs) based on experience from other development teams. If a team already has experience in managing TD, it can identify other commonly used practices or other PARs faced and can also identify enabling activities (enabling practices) that will improve the team's ability to manage TD. In other words, teams can create their own IDEA diagrams.

As a communication device, results also suggest that IDEA diagrams could be used in meetings to discuss TD items, explaining the factors that lead to non-management of TD and presenting possible solutions to minimize the effects of these factors.

For researchers, our findings can guide new investigations on TD management, considering the IDEA diagrams as a starting point. For example, practices and PARs could be further investigated to refine them according to different software contexts. The IDEA diagrams can be automatized as a plugin of an issue tracking tool, such as Jira (https://www.atlassian.com/software/jira) or Asana (https://asana.com/). Lastly, there is still a need of conducting further industrial empirical investigations to evaluate IDEA diagrams as a supporting tool for TD management activities.

6 Threats to Validity

As in any empirical study, there are threats to validity in this work [23]. We attempt to remove them when possible, and mitigate their effects when removal is not possible.

6.1 TAM Study

Construct validity. A threat emerges from the material used to perform the TAM study because the TD items analyzed by the participants can influence their perceptions about the IDEA diagrams. Although we have used actual TD items provided by an industry partner, only the replication of our study with variation in the material can reduce this threat. Another threat arises from the TAM questionnaire due to its questions and length. The participants could misunderstand the questions, and the number of questions could fatigue participants. To mitigate this threat, we conducted two internal validations to identify improvements in study design and its material (questionnaire and training materials). We then piloted the questionnaire before its execution. None of these participants reported issues in answering the questionnaire.

Conclusion validity. The primary threat is that the participants were not allowed to participate in the software project that provided the TD items used in the study nor to talk to the project members. Therefore, it can affect the analysis of practices and PARs conducted by the participants. We assumed this threat as a limitation of the study. As we are not evaluating the final list of practices and PARs, but the use of the diagrams, the participants are able to simulate the work of identification of TD management practices and PARs based on the description of each TD item. Besides, as we did not obtain 100% agreement on the statements of the self-predicted future use construct, we believe that the participants were able to analyze the effort in performing the tasks with and without the diagrams.

External validity. A threat arises from the fact that the study participants were chosen by convenience and were all students (some with industry experience in software development). Thus, although the results are not as generalizable as they could have been with a more representative sample, they provide initial evidence on the investigated topic.

6.2 Interview Study

Construct validity. A threat emerges from the interview script in that the participants could misunderstand its questions. To mitigate this threat, we performed two internal validations and piloted the interviews with two participants with distinct levels of experience. Our goal was to identify the time necessary to run interviews (the mean time was about 30 min) and collect impressions about the questions and improvement points. All was considered fine during the validations and pilot.

Conclusion validity. The primary threat arises from our coding process to analyze the interview transcripts. As this process is subjective, the first author coded the transcripts, and the last author reviewed the extracted codes. These authors conducted a meeting to resolve eventual disagreements.

External validity. A threat arises from the small number of participants that may not be representative of a population. It did not allow us to perform more specific analysis, for example, if different practitioner roles who have different points of view on software projects, also have different perceptions of IDEA diagrams. We assumed this threat as the main limitation of this study. Another threat is related to the fact that study participants were chosen by convenience because we invited only practitioners from our network in the software industry. We decided to use this method because we had truly little control over the availability of subjects, resulting in inviting only practitioners via existing contacts in software organizations. To mitigate this threat, we tried to ensure that our sample was reasonably representative and not strongly biased. For this, we tried to carefully select practitioners from distinct roles, with experience in their roles, and from different organizations.

7 Concluding Remarks

This study presents the assessment of the IDEA diagrams, which provide support for TD prevention, monitoring, and repayment activities. We conduct a TAM study with 72 students enrolled in a software engineering course and an interview study with 11 software practitioners. The results from both studies are positive, complementary, and confirmatory, revealing that the data embedded into the IDEA diagrams and the diagrams themselves are useful for TD prevention, monitoring, and repayment initiatives.

As future work, we intend to: (i) specialize the diagrams considering project variables, such as process mode and company size, (ii) automate the diagrams to provide dynamic manipulation of data, (iii) define how practices and PARs can be collected by software teams to automatically feed the diagram, and (iv) conduct case studies in the software industry to investigate when and how the IDEA diagrams can be used as part of project activities.

References

1. Avgeriou, P., Kruchten, P., Nord, R., Ozkaya, I., Seaman, C.: Reducing friction in software development. IEEE Software 33 (1), 66–73 (2016).
2. Falessi, D., Kazman, R.: Worst smells and their worst reasons. In: Proceedings of the IEEE/ACM International Conference on Technical Debt, pp. 45–54. IEEE, Madrid (2021).
3. Izurieta, C., Vetrò, A., Zazworka, N., Cai, Y., Seaman, C., Shull, F.: Organizing the technical debt landscape. In: Proceedings of the International Workshop on Managing Technical Debt, pp. 23–26. IEEE, Zurich (2012).
4. Seaman, C., Guo, Y., Zazworka, N., Shull, F., Izurieta, C., Cai, Y., Vetrò, A.: Using technical debt data in decision making: potential decision approaches. In: Proceedings of the International Workshop on Managing Technical Debt, pp. 45–48. IEEE, Zurich (2012).
5. Spínola, R., Zazworka, N., Vetrò, A., Shull, F., Seaman, C.: Understanding automated and human based technical debt identification approaches-a two-phase study. Journal of the Brazilian Computer Society 25 (5), 1–21 (2019).
6. Li, Z., Avgeriou, P., Liang, P.: A systematic mapping study on technical debt and its management. Journal of Systems and Software 101, 193–220 (2015).
7. Rios, N., Mendonça, M., Spínola, R.: A tertiary study on technical debt: types, management strategies, research trends, and base information for practitioners. Information and Software Technology 102, 117–145 (2018).
8. Yli-Huumo, J., Maglyas, A., Smolander, K.: How do software development teams manage technical debt? - An empirical study. Journal of Systems and Software 120, 195–218 (2016).

9. Bomfim Jr, M., Santos, V.: Strategies for Reducing Technical Debt in Agile Teams. In: Silva da Silva T., Estácio B., Kroll J., Mantovani Fontana R. (eds) Agile Methods. WBMA 2016. Communications in Computer and Inf. Science, vol. 680, 60–71. Springer, Cham (2017).
10. Behutiye, W., Rodríguez, P., Oivo, M., Tosun, A.: Analyzing the concept of technical debt in the context of agile software development: a systematic literature review. Information and Software Technology 82, 139–158 (2017).
11. Toledo, S., Martini, A., Przybyszewska, A., Sjøberg, D.: Architectural technical debt in microservices: a case study in a large company. In: Proceedings of the International Conference on Technical Debt, pp. 78–87. IEEE, Montreal (2019).
12. Silva, V., Jeronimo Junior, H., Travassos, G.: A taste of the software industry perception of technical debt and its management in Brazil. Journal of Software Engineering Research and Development 7 (1:1), 1–16 (2019).
13. Apa, C., Solari, M., Vallespir, D., Travassos, G.: A taste of the software industry perception of technical debt and its management in uruguay: a survey in software industry. In: Proceedings of the 14th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement, pp. 1–9. ACM, New York (2020).
14. Apa, C., Jeronimo, H., Nascimento, L., Vallespir, D., Travassos, G.: 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 (2020).
15. Rios, N., Mendes, L., Cerdeiral, C., Magalhães, A., Perez, B., Correal, D., Astudillo, H., Seaman, C., Izurieta, C., Santos, G., Spínola, R.: 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 (2020).
16. Aragão, B., Andrade, R., Santos, I., Castro, R., Lelli, V., Darin, T.: TestDCat 3.0: catalog of test debt subtypes and management activities. Software Quality Journal 30, 181–225 (2022).
17. Freire, S., Rios, N., Pérez, B., Castellanos, C., Correal, D., Ramac, R., Tausan, N., Mandic, V., Hernandez, A., López, G., Mendonça, M., Izurieta, C., Falessi, D., Seaman, C., Spinola, R.: Pitfalls and Solutions for Technical Debt Management in Agile Software Projects. IEEE Software 38 (6), 42–49 (2021).
18. Shahir, H., Daneshpajouh, S., Ramsin, R.: Improvement strategies for agile processes: A SWOT analysis approach. In: Proceedings of the International Conference on Software Engineering Research, Management and Applications, pp. 221–228. IEEE, Prague (2008).
19. Rios, N., Spínola, R., Mendonça, M., Seaman, C.: 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. Empir. Softw. Eng. 25, 3216–3287 (2020).
20. Davis, F.: Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly, 319–340 (1989).
21. Seaman, C.: Qualitative methods in empirical studies of software engineering. IEEE Transactions on Software Engineering 25(4), 557–572 (1999).
22. Strauss, A., Corbin, J.: Basics of qualitative research: techniques and procedures for developing grounded theory. Sage Publications, Thousand Oaks, Calif. XIII (1998).
23. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M., Regnell, B., Wesslén, A.: Experimentation in software engineering: An introduction. Springer (2012).