Universidade Federal do Rio Grande do Norte
Centro de Ciências Exatas e da Terra
Departamento de Informática e Matemática Aplicada
Programa de Pós-Graduação em Sistemas e Computação

Business-driven Technical Debt Prioritization

Rodrigo Rebouças de Almeida

Natal-RN
February, 2022

Rodrigo Rebouças de Almeida

Business-driven Technical Debt Prioritization

A thesis submitted to the Graduate Program in Systems and Computing (PPGSC) in conformity with the requirements for the Degree of Doctor of Philosophy

Supervisor

Prof. PhD. Uirá Kulesza

Co-supervisor

Prof. PhD. Christoph Treude

PPgSC – Programa de Pós-Graduação em Sistemas e Computação
DIMAp – Departamento de Informática e Matemática Aplicada
CCET – Centro de Ciências Exatas e da Terra
UFRN – Universidade Federal do Rio Grande do Norte

Natal-RN

February, 2022

Universidade Federal do Rio Grande do Norte - UFRN
Sistema de Bibliotecas - SISBI
Catalogação de Publicação na Fonte. UFRN - Biblioteca Setorial Prof. Ronaldo Xavier de Arruda - CCET

Almeida, Rodrigo Rebouças de.
Business-driven technical debt prioritization / Rodrigo Rebouças de Almeida. - 2022.
134f.: il.

Tese (Doutorado) - Universidade Federal do Rio Grande do Norte, Centro de Ciências Exatas e da Terra, Programa de Pós-Graduação em Sistemas e Computação. Natal, 2022.
Orientador: Prof. Dr. Uirá Kulesza.
Coorientador: Prof. Dr. Christoph Treude.

1. Computação - Tese. 2. Technical debt - Tese. 3. Technical debt management - Tese. 4. Business-driven technical debt prioritization - Tese. 5. Business process management - Tese. 6. Gerência de dívidas técnicas - Tese. I. Kulesza, Uirá. II. Treude, Christoph. III. Título.

RN/UF/CCET CDU 004

Elaborado por Joseneide Ferreira Dantas - CRB-15/324

RODRIGO REBOUÇAS DE ALMEIDA

"*Business-driven Technical Debt Prioritization*"

Esta Tese foi julgada adequada para a obtenção do título de Doutor em Ciência da Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Sistemas e Computação do Departamento de Informática e Matemática Aplicada da Universidade Federal do Rio Grande do Norte.

Uirá Kulesza

Prof. Dr. UIRÁ KULESZA
Presidente

Prof.ª Dr.ª ANNE MAGALY DE PAULA CANUTO
Coordenadora do PPgSC

Banca Examinadora

Carolyn Seaman

Examinador(a) Externo(a): **Dr. CAROLYN SEAMAN**

Christoph Treude

Examinador(a) Externo(a): **Dr. CHRISTOPH TREUDE**

Eduardo Aranha

Examinador(a) Interno(a): **Dr. EDUARDO HENRIQUE DA SILVA ARANHA**

Eiji Adachi

Examinador(a) Externo(a): **Dr. EIJI ADACHI MEDEIROS BARBOSA**

Marcos Kalinowski

Examinador(a) Externo(a): **Dr. MARCOS KALINOWSKI**

Discente: **RODRIGO REBOUÇAS DE ALMEIDA**

Natal, 23 de fevereiro de 2022

I dedicate this work to Ayla, Ana Luísa, and Gabriela.

Acknowledgment

Thank you to Ayla, my wife, and my daughters Ana Luísa and Gabriela for all the support and patience. During all the time, they were my ground and motivation. Having a Ph.D. student at home sometimes is not easy. Also, thank you to my parents Sebastião and Hélia, and my brother and sister, Daniel and Raquel.

A special thank you to Uirá and Christoph for all the support and patience. It was an honor to have them as advisors. Besides guiding me during this journey, they were examples of research partners. Much more than they said, the way they behaved in several situations was my reference. Their guidance was fundamental to raising my capacity as a researcher.

The duty to implement a tool on ongoing research is hard, and it was possible because of the strong developers by my side. Numerous changes, pressure to meet research targets, and numerous discussions made the approach viable to be implemented. Thank you to Rafael Ribeiro, Emerson Ruan, and Robson Alves. They were directly involved in implementing the technical debt management tool (Tracy-TD). Also, thank you to Jackson Anderson for the help in maintaining and evolving Tracy-TD during the last year.

The only way to deeply explore and understand the business perspective is to have experienced professionals involved in the research. I'm thankful for all the practitioners that were part of this research, as interviewees, anonymous respondents, and members of the companies that were partners during this journey.

Thank you to Dataprev, and the involvement of Flavio Ronison, Aliandro Higino, Filipe Teixeira and Rodrigo Almeida. Thank to Conductor Technology, with Bruno Azevedo, Bruno Nóbrega, João Barbosa, Rodrigo Breckenfeld, Rogério

Viana, and Vinicius Machado.

Special thanks to Phoebus Technology in the name of its directors directly involved in this research, Harley Barroso and Anderson Vera. Its product owners and technical leaders André Figueiredo, Carlos Ferreira, Daniel Silveira, Pedro Ivo, and Ricardo Neves. A warm thank you to the "Plataforma Digital" team, its product owner Luiz Moura and squad leader José Cavalcanti Netto for the invaluable insights, experience, and support to the research.

Thank you to Guilherme Germoglio, Gregório Melo, Carlos Lima, Diego Cavalcanti, Walfredo Cirne, Felipe Ribeiro, Teoni Valois, Flávio Barata, Yuri Teixeira, Herval Freire, Maurício Linhares, Jorge Alcoforado, professionals and friends who provided several insights and great discussions about technical debt in the industry.

Also, thank you to all colleagues from the CASE research group. The routine of sharing information, experiences, and discussions

Finally, I would not finish this Ph.D. without mentioning my former advisors, Jacques Sauvé and Antão Moura. Part of who I am as a professional, teacher, and researcher also came from their past guidance.

In everything, then, do to others as you would have them do to you.[1]

Jesus Christ

[1] However, don't forget to put yourself in the other's place.

Business-driven Technical Debt Prioritization

Author: Rodrigo Rebouças de Almeida
Supervisor: Prof. PhD. Uirá Kulesza
Co-supervisor: Prof. PhD. Christoph Treude

Abstract

Technical debt happens when teams take shortcuts on software development to gain short-term benefits at the cost of making future changes more expensive. Previous results show misalignment between the prioritization done by technical professionals and the prioritization expected by business ones. This thesis presents a business-driven approach to prioritizing technical debt. The research is organized into three phases: (i) exploratory - a survey with practitioners, to identify the business causes of technical debt interviews; (ii) concept verification - where the proposed approach was evaluated on a multi-case study; and (iii) - design and evaluation - where a design science research, with the involvement of three companies, was conducted to develop Tracy, an approach for business-driven technical debt prioritization; followed by a multiple case study on two other companies. So far, we have identified business causes and impacts of technical debt; we designed the approach for business-driven technical debt prioritization; after we developed a tool based on the approach, we finally ran a multiple case study on two companies to evaluate the solution. Results show a set of the business causes behind the creation of technical debt; and also that the business-driven prioritization of technical debt can improve the alignment and communication between the technical and business stakeholders. We also identified a set of business factors that may drive the technical debt prioritization. *Keywords*: technical debt, technical debt management, business-driven technical debt prioritization, business process management.

List of Figures

List of Tables

Contents

1 Introduction

Technical debt is a problem in software development and evolution that occurs when teams take a shortcut to gain short-term benefits at the cost of making future changes more expensive or impossible [Kruchten, Nord e Ozkaya 2019]. Like in financial management, when we borrow some money and delay its payment, we have to pay interest, the "same" happens with software. For example, when some feature development, documentation, or test is delayed or postponed to be accomplished in the future, there is often an associated cost. This extra-cost is called "interest" [Guo e Seaman 2011]. The metaphor has been widely used in industry and has been a research topic for more than twenty years. [Cunningham 1992, Avgeriou et al. 2016]

Although the metaphor is used to facilitate the communication between information technology IT and business stakeholders, the state of the art of technical debt still lacks more appropriate treatment of how technical delays affect the business and vice-versa. Seaman and Guo [Seaman e Guo 2011] also argue that a comprehensive technical debt theory that formalizes the relationship between the cost and benefit sides of the technical debt concept must be developed.

1.1 Problem statement

The report from the Dagstuhl Seminar 16162 [Avgeriou et al. 2016] presents a research roadmap for Technical Debt (TD). It involved 33 researchers, practitioners, and tool vendors from academia and industry. They argue that "business value is central to delivering effective mechanisms for managing TD in practice". Also, that "demonstrating the benefits of considering TD in management decisions is a key area for TD researchers". Other previous research work [Buschmann 2011, Lim, Taksande e Seaman 2012, Kruchten et al. 2012, Wolff e Johann 2015] also motivate that research and practice on technical debt must consider the business perspective on TD management activities.

Several secondary [Ampatzoglou et al. 2015, Ribeiro et al. 2016, Fernández-Sánchez et al. 2017, Lenarduzzi et al. 2019, Alfayez et al. 2020] and tertiary [Rios, Neto e Spínola 2018] studies analyze technical debt research. With regard to technical debt prioritization, they report that the criteria, tools, and approaches used to prioritize technical debt lack a business perspective. Lenarduzzi et al. [Lenarduzzi et al. 2019] conducted a systematic literature review on technical debt prioritization and identified only three papers [Martini e Bosch 2015] [Yli-Huumo, Maglyas e Smolander 2016] [Gupta et al. 2016] that use business-related constraints. They highlight that based on most surveys conducted with practitioners, customer and business factors are the most important to consider when prioritizing technical debt. However, only a few studies addressed such factors.

In this context, the problem addressed in this thesis is to support technical debt prioritization using the business perspective. In this way, we analyze the business causes behind the creation of technical debt and propose a business-driven technical prioritization approach.

1.2 Thesis Proposal

This thesis aims to answer the following general research question: **How does the business perspective influence on the prioritization of technical debt?**

To answer this question, we developed four studies, summarized below. First, to understand the relationship between business concerns and technical debt, we asked practitioners how business aspects contribute to technical debt. After, we proposed and ran the introductory approach towards a business-driven technical debt prioritization. After that, we refined and evaluated the approach to business-driven TD prioritization and implemented a tool to support it. Finally, we run another multiple case study with two companies to evaluate the approach in an industrial scenario. Each study was driven by its specific research questions.

1.2.1 Study 1 - Beyond tight deadlines: what are the business causes of technical debt?

We run a survey with 71 anonymous participants from companies in 18 industries across ten countries in this study. Our objective was to answer the following research question: *RQ 1.1. How do business decisions contribute to technical debt?*

The results show a set of the business causes for technical debt and a cause-effect model that relates the various business causes of technical debt to each other and explains their impact on technical debt.

1.2.2 Study 2 - Aligning Technical Debt Prioritization with Business Objectives: A Multiple-Case Study

In this study we report on a multiple-case study of how two large software development companies handle technical debt items, and we show how taking the business perspective into account can improve the decision making for the prioritization of technical debt. We also propose a first step towards an approach that uses business process management (BPM) to manage technical debt. We interviewed a set of IT business stakeholders, and we collected and analyzed different sets of technical debt items, comparing how these items would be prioritized using a purely technical versus a business-oriented approach.

In this study we answered: *RQ 2.1. How can the business perspective influence the prioritization of technical debt? RQ 2.2. Does the business perspective captured through business process management facilitate the prioritization of technical debt?*

1.2.3 Study 3 - Tracy: A Business-driven Technical Debt Prioritization Framework

In this study, we designed and evaluated a business-driven approach to prioritize technical debt items. The research is organized into four phases: (i) exploratory - to identify the research focus; (ii) concept verification - where the proposed approach was evaluated on a multi-case study; (iii) solution - where a design science research [Wieringa 2014] was conducted to develop Tracy, a framework for technical debt prioritization; and (iv) evaluation - where the approach was applied to industry scenarios. Results so far show that the business-driven prioritization of technical debt items can improve the alignment and communication between the technical and business stakeholders.

The designed approach followed the design goal/problem statement: *Improve technical debt prioritization by designing a business-oriented decision-making framework to promote the alignment between technical decisions and business expectations.*

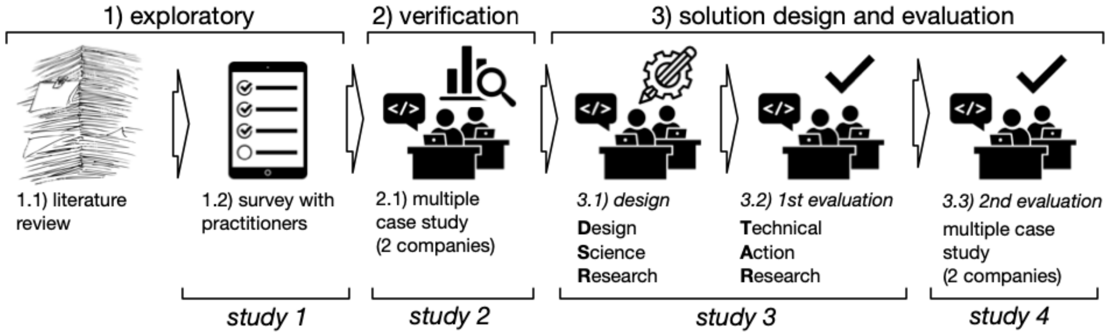


Figure 1: Thesis overview.

1.2.4 Study 4 - Business-driven Technical Debt Prioritization: a Multiple Case Study

In this fourth study, we implemented a tool based on the approach and performed an industrial multiple case study to evaluate how the business-driven technical debt prioritization fits a real scenario. This study involved both technical and business stakeholders from two companies. Finally, this last study answered the following research questions: *RQ 4.1: How does the proposed business-driven approach impact the technical debt prioritization? RQ 4.2: What are the business stakeholder's perceptions regarding factors that should influence technical debt prioritization? RQ4.3: What are the benefits and potential improvements of running the approach in a real scenario?*

1.3 Methodology

Figure 1 shows the three phases of this thesis methodology (1) exploratory, 2) concept verification, 3) design and evaluation), and relates them to the four studies discussed in the previous section.

1) Exploratory phase: The objective of this phase was to identify the focus and clarify the research problem. It was composed of the following three activities.

1.1) Literature review: the review of the state of the art to identify the open research problems related to technical debt management. As a result, we identified a lack of studies considering the business perspective to technical debt management decisions.

1.2) Survey with practitioners: In order to understand how business aspects are related to technical debt, we ran a survey with 71 participants from companies in 18 industries across ten countries in this study.

In the next phase (**concept verification**), we verified if a business-driven prioritization would contribute to the prioritization of technical debt.

2) Concept verification: The results from phase one led us to propose a technical debt management approach that evaluates business value through business processes. To evaluate the approach, we conducted a multiple-case study [Yin 2018] in two companies. Results show that the proposed business-driven approach can improve the prioritization of technical debt, considering business expectations [Rebouças de Almeida et al. 2018].

3) Approach design and evaluation: After having initial evidence that the business-driven approach could help the prioritization of technical debt, we moved to the **solution design and evaluation** phase, organized as design and two evaluation steps.

3.1) Approach design: to develop a solution for the business-driven prioritization problem, we conduct design science research (DSR) [Wieringa 2014] with participants from 3 companies.

3.2) 1st evaluation: the next step towards the end of the research is the application of Technical Action Research [Wieringa 2014] to evaluate if the solution meets the design requirements. We executed an initial evaluation which shows that the developed framework is coherent in its structure [Rebouças de Almeida, Treude e Kulesza 2019].

3.3) 2nd evaluation: at the last step of the evaluation phase, we put the solution on the ground to be evaluated in a multiple industrial case study. The study was instrumented by Tracy-TD, a TDM tool developed in this research that allowed us to handle real data and support real technical debt prioritization decision-making.

1.4 Thesis Organization

This thesis is organized as follows: Chapter 2 presents the survey results about business causes for technical debt. Chapter 3 present a multiple case study to verify the early approach to prioritize technical debt. Chapter 4 presents the business-driven prioritization approach and its first evaluation. Chapter 5 presents the industrial evaluation through a multiple case study. Chapter 6 presents our conclusions and include the next steps.

2 Beyond tight deadlines: what are the business causes of technical debt?

2.1 Introduction

Business aspects have been identified as a significant force behind the creation of technical debt [Ampatzoglou et al. 2015, Kruchten, Nord e Ozkaya 2019]. Technical debt has been studied from many perspectives, from code to human behavior aspects [Rios, Neto e Spínola 2018]. However, the business dimension lacks deeper exploration. Causes of technical debt such as "tight deadlines" and "business pressure" occupy the top ranks among causes of technical debt [Kruchten, Nord e Ozkaya 2019, Rios et al. 2020], but what is behind the "business pressure"? What are the business causes behind tight deadlines? What drives the prioritization of features that pushes quality matters to the back burner?

We conducted a survey to understand better these forces and the practitioners' perspectives on how business decisions contribute to technical debt.

This survey complements our business-driven technical debt prioritization approach [Rebouças de Almeida et al. 2018, Rebouças de Almeida et al. 2021] where we found that the business perspective can make a relevant contribution to technical debt management.

2.2 Method

The research question we answer in this chapter is: **How do business decisions contribute to technical debt?** Here, we focus on the answers to two of the survey's questions:

- Q1: "To what extent do business decisions lead to the creation of technical debt?", a closed question with five answer options: "not at all," "to a very small extent," "to some extent," "to a great extent" and "to a very great extent."
- Q2: "Could you give examples of how business may contribute to technical debt?", an open-ended question.

The survey was primarily publicized on social networks (LinkedIn, Twitter, and Facebook) and via snowballing (i.e., respondents forwarding the survey to other potential respondents). We received 71 anonymous and valid responses. The respondents were aware of the TD concept (92%) and could give concrete examples of technical debt (100%) after being shown a definition.

The majority of respondents indicated having more than ten years of experience (63%) and primarily having technical responsibilities (66%), while a significant minority (25%) indicated both technical and business responsibilities, and 9% pure-business responsibilities. Most of the respondents work for large companies (more than 1,000 employees), in a diverse range of industries including software (31%), government (13%), and finance (8%). Respondents were located in Brazil (59%), North America (25%), and Europe (14%).

We also asked the respondents to give examples of how business may contribute to technical debt. We coded the responses and identified 12 causes divided into three categories: pure business, business/IT gap, and management. The responses contained 1644 words (median: 23.5, standard deviation: 23.2).

2.3 Results

When asked to what extent business decisions lead to the creation of technical debt for the first question (Q1), 96% of the respondents indicated that business decisions lead to the creation of technical debt (to some extent: 23%; to a great extent: 51%; to a very great extent: 23%) while only 4% indicated no or low influence.

Based on analyzing the survey responses (Q2), we classified the business causes of technical debt into three categories using thematic analysis [Braun e Clarke 2006]. One author coded the answers, and the other two reviewed them. The categories that emerged from the analysis are: **pure-business** (i.e., those related to business decision-making and external market forces); causes related to the **business/IT gap** (i.e., knowledge and planning gap); and **management**.

In addition, based on co-occurring mentions in the survey and explicit mentions of cause/effect relationships, we could identify business causes of *tight deadlines* and *feature over quality*, the most cited management causes for technical debt by our respondents. For example, one participant answered that "[Business deadlines] may press towards [fulfilling requirements as soon as possible] because of [competition]." From this answer, we could relate the codes *time to market* (from "business deadlines") and *rush to deliver to beat competitors*, as a cause of *feature prioritization.*

2.4 Business causes of technical debt

The **management** category (56 codes) included well-known causes of technical debt [Rios et al. 2020, Freire et al. 2021] like *tight deadlines* (19) and the problem of prioritizing *features over quality* (17). Both are aspects that usually are related to short-term benefits. Causes like *bad requirement elicitation* (11) and *changes of development scope within development sprint* (5) were also mentioned. Since the management causes are well discussed [Rios et al. 2020], here we focus on the **business** and **business/IT gap** causes behind the two most cited management causes of technical debt: **tight deadlines** and the prioritization of **feature over quality**.

Tight deadlines are commonly identified as the top management cause of technical debt [Rios et al. 2020]. With no time left to deliver features, teams must postpone activities in order to meet target releases. Tight deadlines were also the most cited cause of technical debt in our survey. Besides being a cause of technical debt, *tight deadlines* are a consequence of many other problems related to **pure business** and the **business/IT gap**. 30% of the respondents mentioned "tight deadlines" as a consequence of other problems.

After *tight deadlines*, *feature over quality* was the second most cited management cause of technical debt. The prioritization of features is often driven by business pressure, like the value perception. For example, "features create value," a business respondent argues that "A team can invest a week into (i) a new feature that will make 50 million revenue over a year or (ii) can use the same time to make their framework more robust for running regression tests. If the team invests in (ii), that will reflect on the company's quarter results negatively, thus pulling shares down. The team is pressured by finance to put all of its effort in (i)."

While the business impact in the context of technical debt is sometimes reduced to tight deadlines [Rios et al. 2020], our analysis reveals a much more complex picture of how external forces and gaps between domains play significant roles in the creation of technical debt.

There are business causes for technical debt that cannot be avoided, e.g., a business opportunity or a customer's demand, but a subset of business pressures can be reduced if well managed. Our results provide a set of causes behind the two

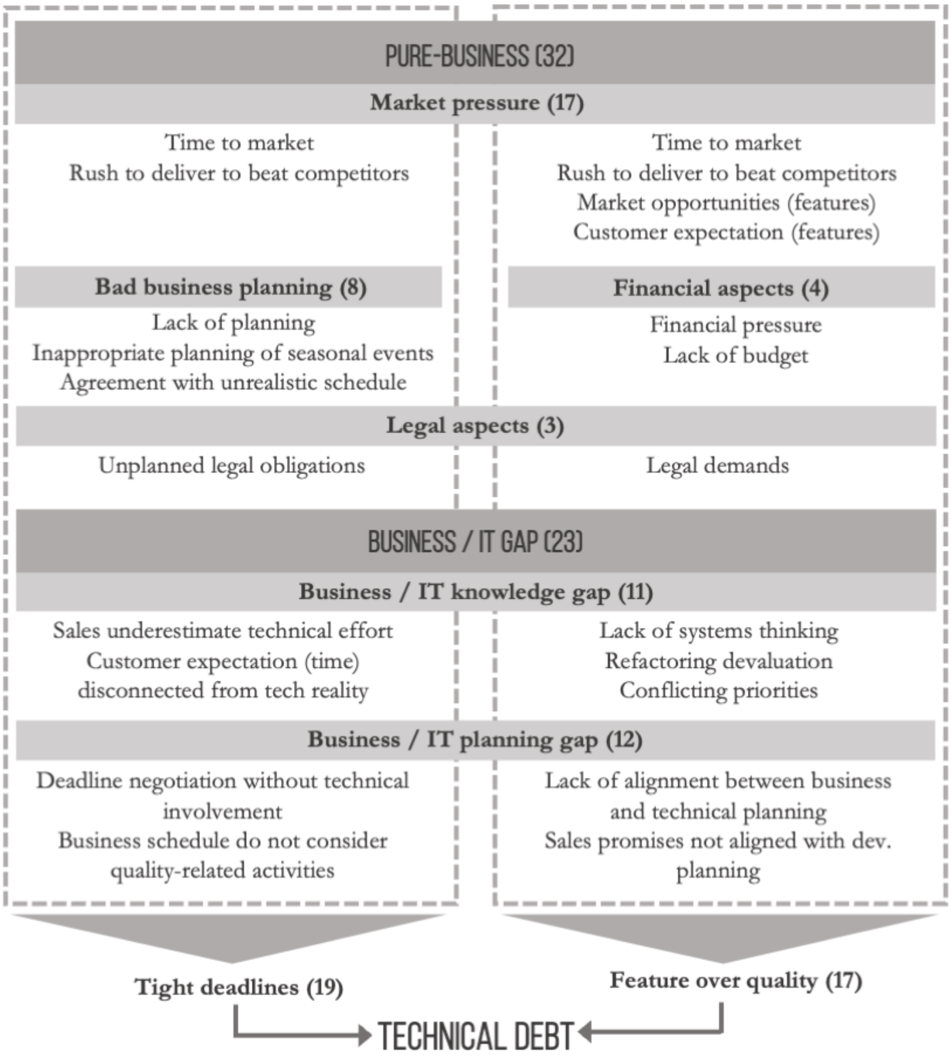


Figure 2: Business causes of Technical Debt, with details for the causes of *tight deadlines* and *feature over quality*

leading management causes for technical debt "tight deadlines" and "feature over quality" prioritization. **When we better understand the business causes of technical debt, we can identify problems that could be avoided, thus preventing the creation of technical debt in the first place.**

Figure 2 presents the code categories of causes of technical debt classified into two main groups: **Pure-business** (32) and **Business / IT gap** (23). It also presents the particular causes (the two descending dashed arrows) for **tight deadlines** (19) and **feature over quality** (17). For example, **financial aspects** (4) is a general cause of technical debt, and *financial pressure* plus *lack of budget* were mentioned as financial causes of the prioritization of **feature over quality**.

In the following, we present the code categories, the number of mentions in the survey and representative quotes from our respondents.

2.4.1 Pure-business

Pure business aspects are the technical debt causes linked to problems from the business side, like marketing pressure, financial aspects, business planning, legal and political aspects. When the business stakeholders and the client rush for new features and prioritize features over quality, this directly impacts the development schedule. The time to market to beat competitors with new products and features is another point of pressure on development deadlines.

2.4.1.1 Market pressure

The most frequently mentioned pure-business cause for technical debt was market pressure (17 mentions). The market pressure is caused by customers, competitors, opportunities, and time to market.

Time to market may cause technical debt by creating forces to release features in a rush to beat the competitors. These forces are causes of *tight deadlines*, and *feature over quality*, like one respondent wrote: "Releasing features before your competitors may give you business advantages. That could motivate tight deadlines and technical debt." This "rush" to deliver may occur in prototypes planned to be delivered as a production-ready solution, causing *feature over quality* and *tight deadlines*, since the planning does not consider the complete set of features and the quality aspects that should be considered in a production-ready solution. Then, the prototype is shipped as a product, with technical debt.

Market opportunities and *customer expectations* regarding features to be delivered also play a role in prioritizing *feature over quality* aspects. It is essential to highlight that these are normal and even expected business pressures. However, sometimes new customer demands are delivered to development teams as "urgent", without proper prioritization and expectation management. One respondent said that "Acquisition of new markets, growing the reach of the company/product, which usually leads to a larger cash flow into the company is usually a lot nicer on the eyes of stakeholders than house maintenance, which tends to de-prioritize projects aware of such problems and usually only when shit really hits the fan or things slow down a lot, that's when people review priorities and we end up getting to take time to clean things up."

Inappropriate management of customer expectations may create unnecessary pressure to deliver a product or service. Arguments like "the customer wants to see the first version / MVP (Minimum Viable Product) ASAP" force teams to focus on creating pressure on the delivery of features and postpone work towards quality attributes, thus creating technical debt.

2.4.1.2 Bad business planning

The second most cited pure-business cause for technical debt was *bad business planning* (8). Arbitrary deadlines, bad agenda planning (e.g. *lack of long-term planning for features and projects*), lack of anticipation for recurring business events, frequent "urgent" features without previous planning and hard and difficult-to-negotiate deadlines are some of the mentioned problems that lead to so-called "business pressure." *Inappropriate planning of seasonal events* also creates pressures on deadlines that could be avoided. For example, "for seasonal marketing and promotional events that happen on the same date, every year, businesses demand new projects and platforms to deal with the similar problems over and over again. More than ten years of new 'urgent' features coming top-down from business with no planning and no care about tests."

2.4.1.3 Financial aspects

Financial pressure also influences technical debt when there is *lack of budget* to address quality and non-functional requirements (*feature over quality*). As for "lack of budgeting for full implementation of solutions", sometimes the market demands lots of changes, new features, and prototypes, but does not budget for full implementation.

2.4.1.4 Legal aspects

Legal aspects also play a role in creating technical debt since it is an external force that can demand new features under unexpected deadlines. They affect the business schedule and prioritize features (e.g., the new European GDPR privacy law created external demand for systems and services). In our survey, one respondent said that sometimes business planning neglects *legal demands* and schedules, "often the business demands legal requirements after a legislation term has already expired, creating urgency for the software development." Also, the "lack of alignment between *legal obligations* and the time to demand the changes affected by legal implications" can put pressure on deadlines and feature prioritization.

2.4.2 Business/IT Gap

Besides the business pressure discussed in the previous section, the gap between business and IT is another cause of technical debt. We categorized the gap into two dimensions: knowledge and planning. The *Business / IT Knowledge gap* encompasses the technical debt causes related to the lack of knowledge from one area about the other. *Business / IT planning gap* is a category of technical debt causes related to deadlines, schedules, planning, and similar concerns.

2.4.2.1 Business / IT Knowledge Gap

Business stakeholders tend to see the development teams as "black boxes." Failing to account for integration concerns, technical impact, and underestimating implementation effort are some of the mentioned causes of technical debt. Furthermore, the *lack of technical involvement in business decision-making* can lead to "bad contracts with service providers/partners, leading to integration workarounds." Finally, this business/IT gap leads business professionals to "create product roadmap[s] with little understanding of technology and organizational limitations."

The knowledge gap between business and IT also contributes to *tight deadlines*. As one respondent stated, "sales and business analysts underestimate implementation effort." For example, one respondent said that "salespeople try to sell more than the company can deliver, sales and business analysts underestimate implementation effort or cut down schedules due to client request...". The *customer expectation* regarding the time when the solution will be delivered is *disconnected from the technical reality*, creating pressure on development deadlines.

The knowledge gap also affects the problem in which features are prioritized over quality aspects. Business stakeholders' *lack of systems thinking* was reported as a cause of feature prioritization, e.g., "rushing to optimize for one part / one group, resulting in negative side effects to the whole / broader organization." Another cause for feature over quality is the *refactoring devaluation*. Some stakeholders do not care about refactoring and other quality aspects, focusing on short-term value delivery. Finally, *conflicting priorities* also contribute to technical debt creation, like "it has to be done fast, it has to be backward compatible, it has to be future proof."

2.4.2.2 Business / IT Planning Gap

The *Business / IT planning gap* received 12 mentions. Problems include the lack of alignment between software requirements and technical development, where business participants make commitments that cannot be handled in the expected time. As a result, *sales promises are not aligned with development planning.*

In this category, the problems with *deadline negotiation without technical involvement* and *the business schedule without considering quality-related activities* were identified as causes of *tight deadlines.* - "Deadlines negotiated on contracts without engineering feedback." In addition, the agenda misalignment occurs when the business planning does not consider the technical planning and vice-versa.

Finally, the lack of alignment between business and technical planning may provoke the prioritization of *features over quality* to deliver value. E.g., "Business usually affects technical debt when commitments are made without consulting the engineering team. This happens because business is always focused on the value being delivered, while teams focus on delivering value AND reducing the cost of maintaining the product."

2.5 Discussion

In this chapter, to complement existing work on the numerous technical and operational causes and consequences of technical debt [Rios, Neto e Spínola 2018, Rios et al. 2020, Verdecchia, Kruchten e Lago 2020, Freire et al. 2021], we focus on the business side of what is causing technical debt.

The relationship between business and technical debt presents itself as an intricate web involving many business aspects from different perspectives. Although we organized the causes of technical debt into two big areas and six categories, they are interrelated. Business aspects added to the business/IT gap and management make a significant contribution to the creation of technical debt.

Market pressures related to customer demands, time to market, and competitors are the leading business causes for technical debt, with the business/IT gap further exacerbating the problem. Tight deadlines were the most cited management cause of technical debt. Going a step further, we uncovered that tight deadlines are caused by a set of pure-business, business/IT gap, and other management causes (see Figure 2). The misalignment between decision-making and planning, and the lack of knowledge about technical and business matters are also relevant causes of technical debt.

It is important to note that many of the presented business-related causes of technical debt cannot be completely avoided. Technical debt provides short-term benefits and incurring debt can be of strategic value, but it must be managed and adequately prioritized to not accumulate over time.

The presented business causes of technical debt, and the most frequent management causes "tight deadlines" and "feature over quality" can guide decision-making and improve business processes to avoid unnecessary technical debt.

Practitioners should review the business processes and the decision-making chain and consider paying attention to managing communication and involvement between business and technical teams regarding planning, scope, and effort estimation. Teams should look for ways to prioritize technical debt considering business metrics and perspectives to align business and technical aspects.

2.5.1 Limitations

The presented results are based on a set of 71 respondents and cannot be treated as generalizable. To address the sample size limitation, the participants are mostly senior practitioners from diverse companies in diverse industries. All codes and categorization were reviewed by at least one author not involved in the coding. There is no distinction between different types of technical debt regarding the presented causes. The causes were declared independent of the debt type.

2.6 Conclusion

In this chapter, we presented a cause-effect model (Figure 2), which relates the various business causes of technical debt to ‘tight deadlines‘ and ‘feature over quality‘ and explains their impact on technical debt. Practitioners of different roles can use this model to understand the influences on technical debt creation, anticipate issues, and work across business and IT to better manage technical debt.

3 Aligning Technical Debt Prioritization with Business Objectives: A Multiple-Case Study

Earlier versions of the work in this chapter appears in the proceedings of the International Conference on Software Maintenance and Evolution (ICSME18) [Rebouças de Almeida et al. 2018]

3.1 Introduction

In this chapter, we present the result of our study where we collected and analyzed a set of 188 technical debt items from two large software development companies and conducted a focus group and interviews with IT and business stakeholders to understand how taking the business perspective into account can improve the decision making related to technical debt prioritization. As a result of our case study, we also propose an extension of Seaman and Guo's framework [Seaman e Guo 2011] by using business process management (BPM) [Dumas Marcello La Rosa e Reijers 2018] to improve the understanding of how critical or urgent a technical debt item is. Our results show evidence that taking business priorities into account can change decisions related to technical debt prioritization. To the best of our knowledge, this is the first work that uses business process management to support technical debt management decision-making.

The main aim of our study is to understand how taking the business perspective into account can affect the prioritization of technical debt items. We set out to answer the following research questions:

- ***RQ 1. How can the business perspective influence the prioritization of technical debt?*** To answer this research question, we interviewed IT and business stakeholders from two large software development companies, and we collected and analyzed a set of 188 technical debt items from different systems, comparing how these items would be prioritized using a purely technical approach versus a business-oriented approach.

- ***RQ 2. Does the business perspective captured through business process management facilitate the prioritization of technical debt?*** We explore how business process management (BPM) can contribute to making technical debt prioritization more aligned with the business objectives. Fourteen different business processes in two cases were identified, and some of them were modeled. These business process models were used to analyze how the information about two business metrics (criticality and urgency) contributes to the technical debt prioritization.

Our results show that using business process management to capture the business perspective facilitates the prioritization of technical debt in order to address business expectations. It also helps to improve the argumentation from the technical side to convince business stakeholders to prioritize what was previously considered pure-technical problems.

3.2 Case Study

The objective of this study was to investigate whether business process management is a viable tool to support technical debt decision making. To answer our research questions, we conducted a multiple-case study [Yin 2018] with two software development companies.

3.2.1 Theoretical basis

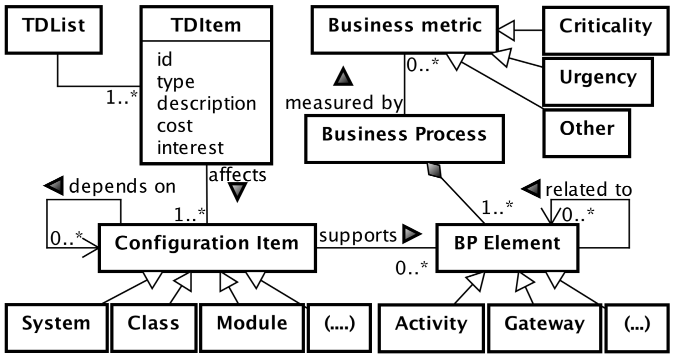


Figure 3: Relationships between the technical debt list and the business process

In this study, we collect the business perspective of technical debt items through their relationship with business processes. Figure 3 shows our conceptual model of how technical debt items and business processes are related to each other. The model shows that a technical debt list “TDList” is related to one or more technical debt items “TDItems” which affect one or more “Configuration Items”. A configuration item can be any technical artifact or system or service which is directly or indirectly affected by technical debt. For example, a test debt item can affect a Java class which can affect a system module, which then can affect an IT service, which can support a business activity, and finally a business process. All items, from the Java class to the IT service, are instances of configuration items. A configuration item can support different business process elements “BP Elements”. Business processes are what companies do to deliver value to customers. For example, a “sales” process in an e-commerce company is the set of activities, decisions, and events that must happen to allow the customer to buy products [Dumas Marcello La Rosa e Reijers 2018]. A BP Element can have its priority and criticality evaluated in business terms. BP Elements compose the "Business Process", which also has its overall priority and criticality. This model extends the conceptual model presented by Rios et al. [Rios et al. 2018] by adding the business process perspective.

3.2.2 Case Study Design

The objective of our case study was to gather data about current technical debt from software development companies, to understand how the systems and services affected by the debt support business processes, and to understand the business processes priorities and if these priorities would affect technical debt prioritization decision making. For the case study, we followed the steps outlined by Runeson, Höst [Runeson e Höst 2008] and Yin [Yin 2018]. First, we planned the case study, designed it, prepared the protocol, collected data and finally, analyzed the results. The following subsections will detail each step.

3.2.2.1 Requirements for the case study

We had the following **requirements for teams to participate in the case study**:

- **Availability:** the team must be available to participate in the research, to give access to data and allow the execution of activities such as interviews, focus groups, and observations. In addition, the company must provide access to pure-business, management, and technical stakeholders.
- **Suffer from technical debt and maintain a list of debt items**: (this was the easiest requirement to meet) the team must understand what technical debt is and maintain a list of technical debt items to be handled by the

team. This requirement was essential to avoid research bias: If the teams had not had an existing list of technical debt items, creating such a list for the purpose of this research could have interfered with the perception of priorities.

- **Be exposed to direct business pressures:** the team must be affected by business stakeholders in their day-to-day work. This requirement excludes teams who work on systems which do not have direct business impact, e.g., teams working on infrastructure.

3.2.2.2 Selected cases

We selected two teams from two companies that were part of a set of eighteen industry partners which collaborate with our research group. In this study, we will refer to the companies as "Company A" and "Company B", to their teams as "Team A" and "Team B", and the cases as "Case A" and "Case B".

Both companies are typical software development companies which develop systems for third-party customers. Company A is part of the government, and Company B is private and provides solutions for credit card processing (private label and co-branded). Company A has more than 600 developers and is responsible for the development and service support of more than 200 different products for different government customers. They handle a country-scale data set. In their case, a product is a set of IT solutions in the scope of a business contract. Their products comprise information systems, mobile systems, data processing, and business intelligence solutions.

Company B has 450 employees, more than 300 different projects, and around 95 different clients. The company is focused on solutions for credit card processing. It processes a mean of 2 million transactions per day, accounting for around 130 million dollars per month.

After selecting the companies, we selected teams suitable for our study. Both selected teams develop large-scale software systems and use a commonly-used technology stack and architecture. Team A develops transaction-intensive information systems and does large-scale data processing. Team B works with large-scale transaction processing systems, on private-label credit card processing.

Team A is composed of 22 professionals with roles such as service support, software developer, software architect, technical leader, system analyst, service manager, and account manager. Team B is composed of 8 professionals, with an agile flavor: software developer, technical leader, test analyst, and product owner.

Both teams use a SCRUM-like development process, they develop systems with high business impact, and are directly affected by business pressures.

Regarding technical debt management, the individuals of both teams understand the term and routinely handle cases of technical debt. Team A developed an internal tool to track and prioritize technical debt items. They also established best practice guidelines for developers, and periodically, there is a team of technical leaders who analyze the code produced by their teams looking for technical debt items.

Team B tracks technical debt items using the task management tool Trello. They use this tool in technical meetings, primarily when the technical debt items are responsible for incidents or are delaying the implementation of features.

3.2.3 Data collection and analysis protocol

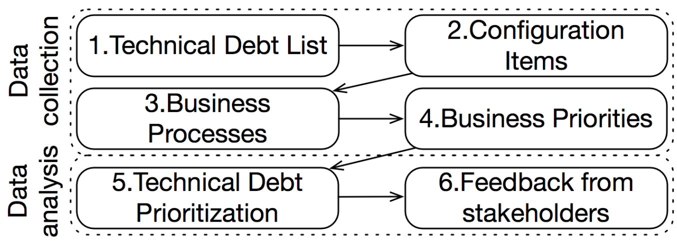


Figure 4: Data collection and analysis main steps

In this section, we present the data collection and analysis protocol applied in both cases. Figure 4 presents the six steps of data collection and analysis. We first collected a list of technical debt items from both teams and worked with the team members to identify the debt items' priorities from a technical point of view as well as the configuration items affected by the debt items. We then worked with business stakeholders from both teams to identify and model the business processes affected by these configuration items and we identified the priorities of the corresponding activities. To prioritize the technical debt items from the point of view of business objectives, we then mapped the prioritized business processes to the list of technical debt items, using the configuration items. This enabled us to compare the technical debt prioritization from both viewpoints: the business perspective and the technical perspective. Finally, we discussed results with stakeholders from both sides. We describe each step in detail in the following.

3.2.3.1 Technical debt list

Case A	technical impact			
technical debts	high	medium	low	total:
low level design	37	13	13	63
code	1	39	4	44
architectural	21	2	-	23
test	14	-	-	14
build	-	-	2	2
security	2	-	-	2
database	2	-	-	2
Total:	77	54	19	150

Table 1: Technical debt categories and impact - Case A

To compare the differences between a technical prioritization and a business-oriented one, we collected a set of technical debt items prioritized according to their impact. **The impact of technical debt is the amount of the consequences of not paying the debt [Guo e Seaman 2011]**. In other words: what happens

Case B technical debts	technical impact high	 medium	 low	 total:
low level design	-	4	-	4
code	-	-	1	1
architectural	-	1	-	1
performance	3	-	-	3
configuration	1	1	1	3
feature	2	1	3	6
security	4	6	-	10
usability	3	5	-	8
database	2	-	-	2
Total:	15	18	5	38

Table 2: Technical debt categories and impact - Case B

if the debt is not paid? To gather this data, we collected a high/medium/low evaluation from technical stakeholders (technical leaders, experienced developers or software architects, for example). We describe the details of this data collection for each case in the following paragraphs.

For Team A, we obtained access to a set of 150 technical debt items with their full descriptions, annotations, and classifications. Table 1 presents a summary of this data. It shows how the technical debt items were ranked regarding their “impact”. Note that this categorization had been made before our study by the technical leaders of Team A who registered the items. We found 19 items with low impact, 54 items with medium impact, and 77 items with high impact. The full technical debt list is available in the companion data [Almeida, Treude e Kulesza].

Team B did not have an explicit list of technical debt items, which required us to analyze a set of 249 user stories and 173 issues from their task management system to select them. After two meetings with a technical leader, we selected 25 technical debt items for the case study. Since the items had not been previously prioritized, we conducted a focus group with seven technical team members (developers, technical leader, and system analyst) with the objective to prioritize the impact of the selected technical debt items from a technical perspective. Each team member classified the technical impact of all items individually at first and subsequently collaboratively discussed any divergences and agreed on the final prioritization.

Table 2 presents a summary of the 38 technical debt items of nine different types. 5 were classified as low impact, 18 as medium and 15 as high.

3.2.3.2 Configuration items

After obtaining the list of technical debt items, we scheduled meetings with technical leaders and senior developers on the two cases to identify which configuration items were affected by each technical debt item.

On Team A, the information about the affected services was identified by the technical leader during code review, for each technical debt item. For example, a particular security debt item had this comment: "Occurrences: PayrollServices.replace, PayrollServices.updatePayrollWithTotalValue, (...)". The comment refers to a Java class which implements a JEE service. We analyzed all technical debt items, identified from the comments all occurrences of each technical debt item and asked the software architect to identify which systems and services were being affected by these occurrences. We conducted three meetings to cover all 150 technical debt items. In the end, they were mapped to five information systems and three batch jobs. A technical leader, with long-time experience on the project, also verified the mapping, adjusting a few mappings and ultimately agreeing on the final result. The companion data [Almeida, Treude e Kulesza] has information about the identified configuration items in Case A.

For Team B, we identified the configuration items affected by each technical debt item by reading the item descriptions and verifying our understanding with a technical leader. Note that the configuration items in Case B were at a higher level (i.e., systems) of abstraction compared to Case A (classes and modules).

3.2.3.3 Business Processes

To understand how the previously identified configuration items support the business, we identified and modeled the business processes supported by the configuration items affected by each technical debt.

In both cases of our study, the team had only used the artifacts related to the description and analysis of business processes in the initial phases of their project (scope definition and high-level analysis). Neither case had structured documentation of the business processes supported by the systems and services, i.e., we had to model the business processes in collaboration with business stakeholders and senior tech leaders.

The business process modeling was done in both teams in two steps. First, we interviewed a system analyst to obtain information about the business processes and modeled them according to Silver [Silver 2011] and Dumas et al. [Dumas Marcello La Rosa e Reijers 2018]. The processes were modeled using BPMN 2.0 [OMG 2011]. A project manager validated them and gave input for adjustments. For each case, the output was a detailed business process model with activities and decisions about internal procedures.

In Case A, technical debt items affect the systems that support a large business process that was detailed in three subprocesses, see Table 3. In Case B, 13 processes were identified and one was detailed, see Table 4.

In Team A, the model was validated by the account manager, in a semi-structured meeting where a researcher presented the business process model and guided the discussion for each business activity. The account manager could make comments and present her concerns about the current modeling.

The process modeled for Team A has three main subprocesses: “Request for Payment”, “Customer Service”, and “Payment” (Table 3). In the process, citizens request a financial benefit (“Request for Payment”), then go to a service center and provide documentation to a “pre-qualification” subprocess, which analyses if they can receive the requested payment. After that, if they are qualified, they are forwarded to be processed in “Service A” (“Forward citizen to Service A”).

The “Request for Service” subprocess is responsible for the processing of an average of more than 1.2 million requests per month. The requests are made using a web application on the Internet. The “Customer Service” subprocess is responsible for an average of 30,000 requests per day, in around 1,500 service centers across the country. The “Payment” subprocess is responsible for the processing of a mean of 700,000 payment orders and handles around US$ 270,000 per week.

In Team B, the models were validated by a senior analyst, who had in-depth knowledge about the business side of the project. The validation meeting was also a semi-structured interview, where each process and activity was revised. In this case, after the meeting, we identified 13 business processes directly affected by the five systems (i.e., configuration items). Table 4 enlists the set of 13 business processes and the 8 activities from the “Invoice payment and scheduling” process.

In this second case, the majority of the business processes could be evaluated as a black box, i.e., without details about activities, events, and decisions. This was possible when a single system or module automated all activities of the process, i.e., if the system or model is affected, the whole process is affected. “5. Card sale” is an example of a highly critical and urgent business process supported by a single system. This was not the case for “Invoice payment and scheduling”, where different activities had different urgency and criticality. Figure 5 shows the activities and decisions from the moment the company schedules a set of payments to be credited to their employees to the moment that the credit is charged to their credit cards.

3.2.3.4 Business Priorities

The next step was to prioritize the business process activities from a business perspective. During semi-structured interviews, we asked business stakeholders to provide their perception of criticality and urgency of the business processes. Some-

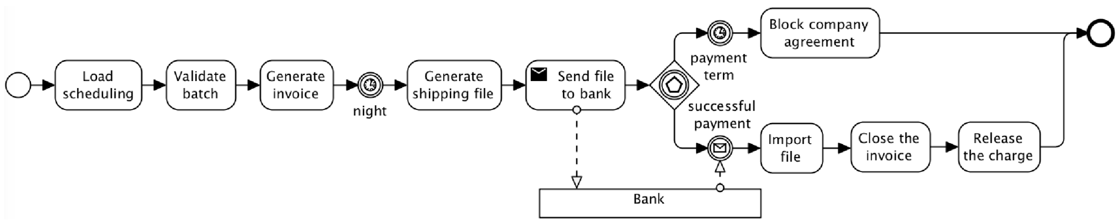


Figure 5: Business process model example: "Invoice payment and scheduling" - Case B

Case A	business	
BP / Subprocess	criticality	urgency
1.Request for payment	low	low
2.Customer Service	medium	high
2.1.Pre-qualification	high	high
2.2.Forward citizen to service A	medium	medium
2.3.Forward citizen to service B	low	low
2.4.Habilitation	high	High
3.Payment	high	medium

Table 3: Business processes and their urgency and criticality - Case A

times they classified the whole business process (in the case where all activities within a process had the same classification), and sometimes they classified specific activities within a process (when different activities had different classifications).

For Case A, we asked the account manager – a business stakeholder – to analyze the modeled business process and evaluate each subprocess. The business criticality was evaluated considering the business value of each subprocess for the citizens while the business urgency was evaluated considering how fast a problem must be solved in order to reduce impact on citizens. The account manager also evaluated the urgency and criticality of the subprocesses of the "Customer Service" process. The final business prioritization is shown in Table 3.

Different from Case A, where the business priorities were a measure of how the process affects citizens, in Case B (see Table 4), the business process priorities were evaluated considering their impact on revenue and the relationship with

Case B	business	
BP / activity	criticality	urgency
1.Anticipation of receivables	high	medium
2.Company agreement registration	low	low
3.Company network registration	medium	low
4.Payment invoice and scheduling	-	-
4.1.Load scheduling	high	medium
4.2.Validate batch	high	medium
4.3.Generate invoice	high	high
4.4.Send file to bank	high	high
4.5.Block company agreement	medium	low
4.6.Import file	high	high
4.7.Close the invoice	high	high
4.8.Release the charge	high	high
5.Card sale	high	high
6.Transfer of money to the network	medium	medium
7.Make reconciliation file available	low	low
8.Manage company agreements	high	high
9.Manage company network	medium	medium
10.Accounting management	low	low
11.Client account management	medium	medium
12.Charge of issuers	high	high
13.Management of the administrator	high	medium

Table 4: Business processes and their urgency and criticality - Case B

business partners. For example, the “Card sale” business process, which enables the customer buying activity, was evaluated as highly critical and highly urgent. “If this process is affected by some problem, the customer can’t use their card”, argues the business analyst. The system which supports this process also has an availability service level agreement (SLA) of 99.98%. The “4.Payment invoice and scheduling” business process has a set of activities which have different criticalities and urgencies. Many of its activities with medium or low urgencies are due to the implementation of automated redundancy or there is a way to run actions manually, e.g., “4.5 Block company agreement”.

3.2.3.5 Technical Debt Prioritization

To prioritize the technical debt items from the point of view of business objectives, we created a new technical debt prioritization, considering the business criticality and urgency. The same procedure was executed in both cases: we mapped the prioritized business processes to the list of technical debt items, using the configuration items. Then we compared the technical debt prioritization from both viewpoints: the business perspective and the technical perspective.

3.2.3.6 Feedback from Stakeholders

After prioritizing the technical debt items using the business perspective, we ran a set of semi-structured meetings with pure-business and technical stakeholders, to discuss the results. All conversations in these meetings and interviews were recorded and summarized into higher-level themes by the author as part of a qualitative analysis. The findings described in the Results section capture these themes.

The meetings had the following structure:

- Show the list of technical debt items and the evaluation of their technical impact. We selected two examples to present in detail. Then we asked if participants understood them and if they had any question about the examples.
- We then presented the technical debt items ordered by their technical impact.
- Next, we showed the list of business processes affected by the technical debt items in the scope of the case study. We asked participants to review the business processes and asked if there is any concern regarding their criticality and urgency ratings.
- Lastly, we presented the prioritization considering the business perspective and compared it with the prioritization using the technical perspective. We

asked if participants had any questions about the new prioritization and we asked if the presented perspective would be useful when handling technical debt items. Finally, we asked them for comments.

In Team A, we ran this meeting three times, one with the account manager (a pure-business stakeholder), one with a software architect, and one with a project manager.

In Team B, we ran this meeting a total of five times: first with the business stakeholder who helped with the business process description and second with a senior developer. We then followed the same meeting structure with 3 additional product owners of 3 different projects from the same company. Since they were from the same company and even though they could not evaluate the accuracy of the technical and business evaluation, they understood the problem and the proposed solution and could evaluate the prioritization using a business perspective.

To evaluate how business and technical stakeholders would use the results from the case study to decide which technical debt items should be selected and how these items should be prioritized in a conflict scenario between business and technical interests, we conducted an additional focus group in Team B. One pure-business stakeholder and one senior developer participated in this focus group. Both stakeholders had more than ten years of experience and had worked with the business model for more than four years.

The focus group was divided into two rounds. In the first round, the participants had access to the 38 technical debt items (each technical debt item had information about its technical impact and its business criticality and urgency); and both participants were asked to select and prioritize ten debt items to be the scope of development in the following development sprints.

In the second round, they were asked to consider their 10 (a total of 20) debt items and negotiate to choose which 10 would be part of the final selection. After the first round, only one technical debt item was selected by both the business and technical participant. After their negotiation, they identified the final selection and prioritization as shown in Table 8. Note that in Case B due to the nature of their debt items, we treated technical debt items affecting multiple business processes as separate items. For example, a highly generic debt item such as "We need a security solution" was broken up into the need for a security solution for System A, for System B, etc. The list of the selected technical debt items is available in the companion data [Almeida, Treude e Kulesza].

3.3 Results

In this section, we present and discuss the answers to our research questions.

3.3.1 RQ 1. How can the business perspective influence the prioritization of technical debt?

With the business prioritization for each process, subprocess and activities in hand and all technical debt items linked to their corresponding business entities (processes, activities, and so on), we step forward to the new technical debt prioritization, considering the business perspective.

Table 7 shows a subset of technical debt items from Case A, each with technical impact, business criticality, and urgency rankings. The table shows the items ordered by their criticality, with higher criticality first. Note the differences and conflicts between technical and business perspectives.

Tables 5 (Case A) and 6 (Case B) show the percentage of technical debt item priorities which matched the business expectation. They show how misaligned this decision would be with business objectives if the team would prioritize the technical debt considering only a technical perspective.

In Case A (Table 5), regarding business criticality, 65% of the technical debt items classified as high priority matched the business expectation. The same applies

business criticality	technical impact high	medium	low
high	**65,1%**	30,1%	4,8%
medium	47,8%	**34,8%**	17,4%
low	27,3%	47,7%	**25,0%**
urgency	high	medium	low
high	**57,1%**	31,4%	11,4%
medium	63,4%	**31,0%**	5,6%
low	27,3%	47,7%	**25,0%**

Table 5: Technical impact versus business impact - Case A

to 34.8% of the medium priority items and 25% of the low priority items. In total, the technical prioritization matched only 48.7% of the criticality prioritization and only 35% matched the urgency expectation. **This result provides evidence on how different a purely technical prioritization could turn out if it had been conducted from a business perspective.**

In Case A, for example, 34.8% (30.1% medium + 4.8% low) of the technical debt items which affect highly critical business processes would not be classified as high priority. Instead, 27.3% of the high impact technical debt items, which affect non-critical business processes, would be prioritized. If we consider the urgency to solve problems on business processes, also in Case A, the situation would be worse, since 42.8% (31.4% medium + 11.4% low) of the technical debt items which affect business processes with high urgency would not be prioritized.

In Case B, (Table 6), we can see that 87.5% of the debt items ranked as medium and high affect business processes with low criticality, while 52% of the debt items that affect highly critical business processes are not ranked as having a high technical impact.

It is clear that we would not expect a complete correspondence between the

business criticality	technical impact high	medium	low
high	**47,1%**	41,2%	11,8%
medium	23,1%	**61,5%**	15,4%
low	50,0%	37,5%	**12,5%**
urgency	high	medium	low
high	**42,9%**	42,9%	14,3%
medium	33,3%	**53,3%**	13,3%
low	44,4%	44,4%	**11,1%**

Table 6: Technical impact versus business impact - Case B

technical and the business perspectives, since the technical aspects which guide the prioritization are different from the business aspects which guide business prioritization. However, the results show that a business-driven prioritization, through the business process perspective, can be useful to support the prioritization of technical debt.

When we showed these results to both business and IT stakeholders in Case A, they understood that something was missing in what was being considered in their prioritizations. The result does not mean that we should consider a purely business-focused perspective when prioritizing a technical debt item, nor a purely technical one. We should consider the trade-offs of each situation to find a balance to enable efficient decision making.

In Case B, the team members mentioned an opportunity to expand the metrics from the high/medium/low ranking to a financial metric in the future. They also saw opportunities to help with scope negotiation with their customers, to convince them to manage technical debt.

The cases where we have a low expectation from the business perspective and a high or medium technical priority (75% for criticality and urgency) may be a

#	description	type	technical impact	business criticality	business urgency
6.6	OO design, lack of information hiding	design	high	high	medium
34.1	Lack of best practice on persistence code	design	low	high	medium
23.1	Problem in EJB annotation	code	low	medium	high
20.1	Wrong class responsibility OO design	design	high	low	low
19.1	Problem in database constraint	database	high	low	low

Table 7: Example of technical debt items ordered by their business criticality - Case A

source of overestimation of a technical issue. For example, item #20.1 in Table 7 refers to a low-level design issue, which affects the system maintainability, classified as medium cost and high interest. This item affects a system which supports the "Request for Payment" subprocess, with a low criticality and urgency, from the business perspective. As a result, solving this item could be delayed compared to the item #23.1, with low interest, in technical terms. Item #23.1, in Table 7, describes a simple annotation problem, at the code level, which is easy to solve and could have low interest. But, since it affects systems which support the Payment business process, it would have a higher priority.

3.3.2 RQ 2. Does the business perspective captured through business process management facilitate the prioritization of technical debt?

We presented the business process modeling together with the prioritization of technical debt items to two business and two IT stakeholders (Case A). The IT stakeholders declared that the business process visualization was useful to support technical debt prioritization. They also argued that *"many times a critical technical debt must be prioritized even if it affects a low critical business process, to reduce the problem of accumulating debt"*. Indeed, the business prioritization is not a silver bullet to define technical debt prioritization, but it provides an important perspective to help in decision making.

#	description	type	technical impact	affected BP.activty	business criticality	business urgency
9	Performance problem on credit limit validation	performance	↑ high	4.8	↑ high	↑ high
10	Performance problem on credit limit validation	performance	↑ high	4.1	↑ high	● medium
11	Performance problem on credit limit validation	performance	↑ high	4.2	↑ high	● medium
37	Lack of captcha on login interfaces	security	● medium	all	↑ high	↑ high
6	Token synchronization	security	● medium	11	● medium	● medium
5	Token synchronization	security	● medium	10	↓ low	↓ low
7	Token synchronization	security	● medium	8	↑ high	↑ high
8	Token synchronization	security	● medium	9	● medium	● medium
34	Enhance visualization on customer invoice	usability	● medium	11	● medium	● medium
35	Enhance visualization on customer invoice	usability	● medium	13	↑ high	● medium

Table 8: Final prioritization after the discussion between business and IT stakeholders - Case B

The IT stakeholders also argued that "*sometimes it is difficult to convince business stakeholders about the risk of acquiring debt, to meet a proposed tight business schedule*". "*It is easier to argue with them that it is necessary to solve a low-level design issue* (as described in Table 7 item #6.6), *since it is explicit that it affects a critical business process*". With a common language and a proper relationship between business processes and technical debt items to handle technical debt management, the communication between the development team and the business stakeholders can be facilitated.

The final focus group, run in Case B with one business and one technical stakeholder (section 3.2.3.6), showed the influence of the business perspective on decision making. Table 8 shows the resulting prioritization. Four technical debt items which affect all business processes were selected. Both focus group participants used their knowledge about business processes as basis for their argumentation. They tried to convince each other by explaining why a particular technical debt item should be selected. Twice, the business stakeholder had to explain details about the business procedures to convince the technical stakeholder. In the end, all selected technical debt items affected a high or medium business criticality and urgency, and the participants selected high and medium technical impacts. The exception was technical debt item 5, which affects a low priority business process, but participants argued that the effort to replicate the "Token synchronization" solution would be low and the gain to pay the debt justified the decision. Another insight was that the business stakeholder used only the information about the business impact and its criticality to make his decisions, i.e., the business perspective captured using business processes was a good basis for decision making.

Finally, in Case A, the account manager explained that using the business perspective for prioritizing technical debt could also provide an objective way to define policies regarding technical debt. Besides the prioritization activity, this can help in decision making about the creation of debt items. Depending on the business process criticality, it would be feasible to deny any high or medium impact debt on it. Her argument leads us to sketch a new approach to prioritize technical debt items based on the business perspective using business process modeling. The next section introduces this approach.

3.4 Discussion and Proposed Approach

This section discusses the findings from the case study which illustrate how aspects of technical debt management and business process modeling can be accomplished in practice.

3.4.1 The tension between technical and business perspectives

The definition of the technical debt metaphor was consistent among the participants on both cases. The examples they provided showed that they had a solid understanding of the theme. The different roles offered varied perspectives about the sources of technical debt. For example, while the senior developer described issues related to low-level coding and developer behavior, the service manager focused on high-level debt, such as architectural debt and infrastructure debt. All participants were unanimous about the interference of business priorities as a source of technical debt. Tight schedules and service incidents were also identified as sources of technical debt.

The technical leaders and software architects are responsible for estimating the impact of technical debt. They have in-depth technical knowledge and experience and a broad perception of the technologies, dependencies, and requirements. Impact is a perception of the debt's dependencies, integration, technical requirements, the probability of resulting in an error or affecting availability, performance, and so on. In summary, all of these aspects are directly and indirectly related to the question, *"What bad things can happen if I do not pay this debt?"*, a technical leader said. She also said that *"it is difficult to evaluate it objectively. Sometimes a line of code can have high impact, and a whole system or module can have low impact. Sometimes we even have to consider the personality of the colleague responsible for a system we depend on. Technically the solution may be straightforward, but aspects such as organization hierarchy and departmental relationships, for example, can greatly increase the impact of debt"*.

From the business side, the account manager from Case A explained that *"sometimes the level of detail from one side and the lack of proper understanding from the other can influence the decision about how urgent or critical a technical debt item is"*. For example, sometimes the technical team discusses a problem at a low level and presents the problem in terms that the business stakeholders cannot understand. *"Sometimes, when the argumentation is lost, the tech guys still come up with some security trouble to convince everyone"*, the business analyst commented. She also pointed out that *"on the other hand, sometimes business stakeholders use 'obscure' motivations to justify a tight schedule or to impose that a feature is more important than solving a technical issue which must wait"*.

A business stakeholder from Case B identified the lack of a broader view of the business as an important problem. She said that *"nowadays people are getting very specialized in their areas and it is difficult to get an overall perspective of the business"*. *"It is quite common to define a scope of a system with one business* *area, and when we start delivering the releases, some conflicts arise with another business area".*

3.4.2 Proposed approach

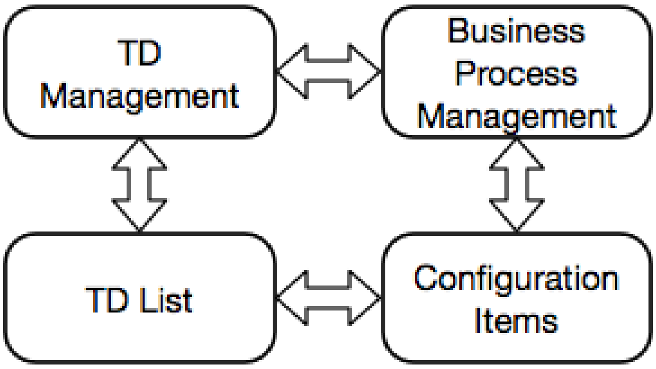


Figure 6: Proposed approach components.

As a result of answering our two research questions, we outline a preliminary conceptual approach which can contribute to technical debt management using the business perspective. The approach extends existing research work by Guo and Seaman [Guo et al. 2011, Guo e Seaman 2011]. Figure 6 shows an overview of its components. Now, besides the technical debt management centered on the technical debt list, there are two new areas: business process management and configuration items. The business process management involves the complete lifecycle of the business processes and a set of management tools to deal with strategic, tactical, and operational aspects of the business perspective.

To apply the approach to technical debt prioritization, it is necessary to:

1. Keep track of a technical debt list;
2. Relate the technical debt items to software and/or infrastructure configuration items;

3. Model the business processes which are supported by the configuration items – technical artifacts of the systems;

4. For each business activity: identify which business aspects contribute to decision making (criticality, urgency, financial aspects, etc.);

5. For each business activity: prioritize the activities considering business objectives;

6. Conduct the technical debt prioritization, using the business perspective.

3.5 Limitations

While our case studies found evidence that it is possible to align technical debt prioritization with business objectives through business process modeling and mapping of technical debt items to business processes, we cannot generalize our findings to other cases. Other companies might have different characteristics and different processes. Naturally, the number of participants we were able to talk to is also limited. However, we note that twenty-two senior professionals participated in this study, on both cases (twelve on Case A and ten on Case B) and that these individuals played different roles in the company (e.g., senior developers, architects, and business stakeholders). Many of the professionals had previous experience with other companies, giving them a broad perspective on the decisions and opinions.

The companies at which this multiple-case study was conducted employ together more than 1000 developers and build complex systems which affect many people and private companies. The approach used to dealing with technical debt by the participant teams is mature, e.g., both teams use specific tools for the management of technical debt. We are therefore optimistic that these cases can be applied to other teams which have a direct business impact.

Our multiple-case study shows evidence that the information about business priorities can change the way companies make technical debt management decisions. In this study, we focus on one technical debt management activity: prioritization. However, there are opportunities to explore other technical debt management activities in the future and their interplay with the business perspective, such as identification, measurement, monitoring, and communication [Rios et al. 2018]. There are also opportunities to explore the business process management, by considering other levels of business decision making, such as operational, tactical, and strategic, and other phases of the business process management lifecycle.

3.6 Related Work

Recent research on technical debt has pointed out the lack of a proper business treatment for technical debt management activities [Martini, Besker e Bosch 2018, Alves et al. 2016, Fernández-Sánchez et al. 2017].

Guo and Seaman [Guo et al. 2011] performed a case study on the release planning of a software application for mobile platforms. They propose a technical debt management framework which considers the principal, the interest amount, and the interest probability when dealing with the technical debt analysis. The scenario consists of an analysis of decision making in the release planning, where a change should be done at a certain point in time and is delayed due to "time-to-market" reasons. The work calculates the probability of the interest by asking experts and the implementation effort is measured in staff-hours. The results show that the use of a technical debt management approach could change key decisions in release management and could avoid the negative effects of the debt. Despite the cost related to the software development team, the business value of the two decisions presented in the work could generate more value compared to the cost incurred through the negative impact on the software side.

There are also researchers who consider business metrics and business values to deal with technical debt. Yli-Huumo et al. [Yli-Huumo et al. 2015], for example, conducted a case study with four companies to understand the relationship between “Business Model Experimentation” and technical debt. Business Model Experimentation is a way to perform business model innovation. This approach is based on the Lean model, promoting business model changes in short life cycles. The authors also argue, based on a literature review, that the relationship between technical debt and business models is not well-studied and requires more examination. The authors performed semi-structured interviews with practitioners from four companies and found that those who use business model experimentation reduce intentional technical debt. This finding is an insight into how involving business stakeholders in the process of technical debt management could increase business value.

Our work also considers a business perspective to support decision making on technical debt. We focus on the prioritization activity, and – unlike related work – we use business process management to help bridge the gap between the technical perspective and the business perspective. To the best of our knowledge, this is the first work which brings these two disciplines together: technical debt management and business process management.

3.7 Conclusions

This multiple-case study addressed the following research questions: RQ 1. How can the business perspective influence the prioritization of technical debt? RQ 2. Does the business perspective captured through business process management facilitate the prioritization of technical debt?

To address these research questions, we performed a multiple-case study in two large software development companies where we observed how the business perspective can affect the prioritization of technical debt. In particular, we have considered how specific business processes and their respective priorities in terms of business urgency and criticality can change the technically oriented prioritization of debt items. Based on the results of the two cases, we make the following contributions:

- We found that the business perspective can affect the prioritization of technical debt items;
- We found that business processes can facilitate the communication and prioritization of technical debt items;
- We extended the analysis model presented by Rios et al. [Rios et al. 2018] by adding the link between technical debt items and the business processes;
- We extended the work by Guo and Seaman [Guo et al. 2011, Guo e Seaman 2011] to propose a conceptual model to support technical debt management decisions while taking business process management into account.

4 Tracy - A Business-driven Technical Debt Prioritization Approach

Earlier versions of the work in this chapter appears in the proceedings of the International Conference on Software Maintenance and Evolution (ICSME19) [Rebouças de Almeida, Treude e Kulesza 2019]

4.1 Introduction

In this chapter, we present a business-driven technical debt prioritization framework, called "Tracy", that prioritizes technical debt considering how IT assets (IT systems which create business value) support a company's business processes. Tracy uses business metrics to support the decision making and has two major benefits: (1) it encourages different stakeholders to consider and identify the business metrics that support decision making about technical debt, and (2) it provides a prioritization mechanism that has the potential to be applied in different business and development contexts.

The proposed framework was constructed through a Design Science Research

Table 9: Groups (A1, A2, ...) and research activities in each DSR phase. In each pair x-y, x is the maximum number of participants, and y is the number of iterations with the group.

	DSR phase		
company	**1-exploratory**	**2-engineering**	**3-evaluation**
A	A1: 2-7 A2: 1-4 A3: 5-3 A4: 1-2 Observ.: 12-12	A5: 1-1 A6: 3-2	-
B	B1: 6-4 B2: 3-1 B3: 2-1	B4: 3-1	B5: 9-7
C	-	C1: 1-1	-
research activities	- interviews - focus groups - observation	- focus groups	- technical action research

(DSR) [Wieringa 2014], which is usually divided into the three phases of exploration, engineering, and evaluation. The exploration and engineering phases involved the participation of 49 professionals from 12 different groups of three companies.

To the best of our knowledge, this is the first research which proposes a technical debt prioritization framework considering business processes and business metrics. The initial evaluation shows that the presented framework is coherent in its structure and that its results contribute to business-driven decision-making on technical debt prioritization.

4.2 Methodology

Due to the importance of considering business aspects when managing and prioritizing technical debt, we are conducting Design Science Research (DSR) to develop a solution for the following design goal/problem statement [Wieringa 2014]: **Improve technical debt prioritization by designing a business-oriented decision-making framework to promote the alignment between technical**

decisions and business expectations.

The designed solution relies on the analysis of information collected over six months and 22 meetings (interviews and focus groups) with seven different groups in two companies, with engineering involving an additional company. Additionally, the author participated as an observer in 12 events (sprint plannings, sprint reviews, incidents, and decision making), where he was able to witness technical debt creation, identification, payment, and business impact.

To conceive a solution for the design problem which would apply to more than one company, the DSR stakeholders were composed of a set of 14 groups of participants from three companies. The groups included 43 professionals: 10 with pure-business roles, 9 with management roles, 6 with technical leadership roles, and 18 with technical roles. No group was aware of the research activities conducted with other groups, and all management and business professionals had more than ten years of professional experience.

The companies have a typical IT organization, with development teams, operations, and use of cloud infrastructure to deliver their services. Two of them provide solutions to the Fintech industry, and the other is a global software consulting company. None of the companies employed a systematic technical debt management approach, often storing technical debt items as "improvements" in the backlog instead.

In each DSR phase, we conducted a set of research activities with selected groups, focusing on different objectives. The phases define the main objectives and the research focus but do not limit the possibility of improving the results of one phase in another, e.g., while designing the solution in phase 2, our understanding of the problem was refined as a result of ongoing discussions related to the solution design.

Table 9 presents the number of groups and the research activities conducted in each phase. All research activity was captured in research log books and audio recordings.

Exploration Phase In the exploration phase, the objective was to understand the constructs (concepts, relations, rules, and motivations) related to technical debt decision making, considering business and management perspectives. The groups that we worked with in this phase had a majority of business and management participants. We conducted interviews, focus groups, and observations [Given 2008] with eight groups, until we reached saturation of the constructs. We also evaluated the business-driven approach to prioritize technical debt in two case studies [Rebouças de Almeida et al. 2018]. The approach was a first step in terms of the work extent and the level of business impact measurement, and a building block for the solution proposed in the next phase.

Engineering Phase The engineering phase started with a first-version prescriptive framework conceived on top of the information from the previous phase and elements from our previous work [Rebouças de Almeida et al. 2018]. Then, we defined key requirements for the solution and iterated over five groups in three companies (see Table 9, phase 2). The groups in this phase had participants with business/management and technical background since the framework needed the input of both profiles. This phase was conducted using focus groups [Given 2008] where the author observed participants using a version of the solution. After each iteration, the framework was reviewed, improved, and presented to the same group and to another one. The end of this phase was triggered when the groups did not have anything to add to the framework.

Evaluation Phase The objective of the evaluation phase was to verify if the solution met the requirements specified in the previous phase, if it solved the design problem, and if the concepts, relationships, artifacts, and prioritization criteria were valid. The evaluation was conducted using Technical Action Research (TAR) [Wieringa 2014] to enable stakeholders to learn about the effects of the framework in practice.

4.2.1 Design Science Constructs

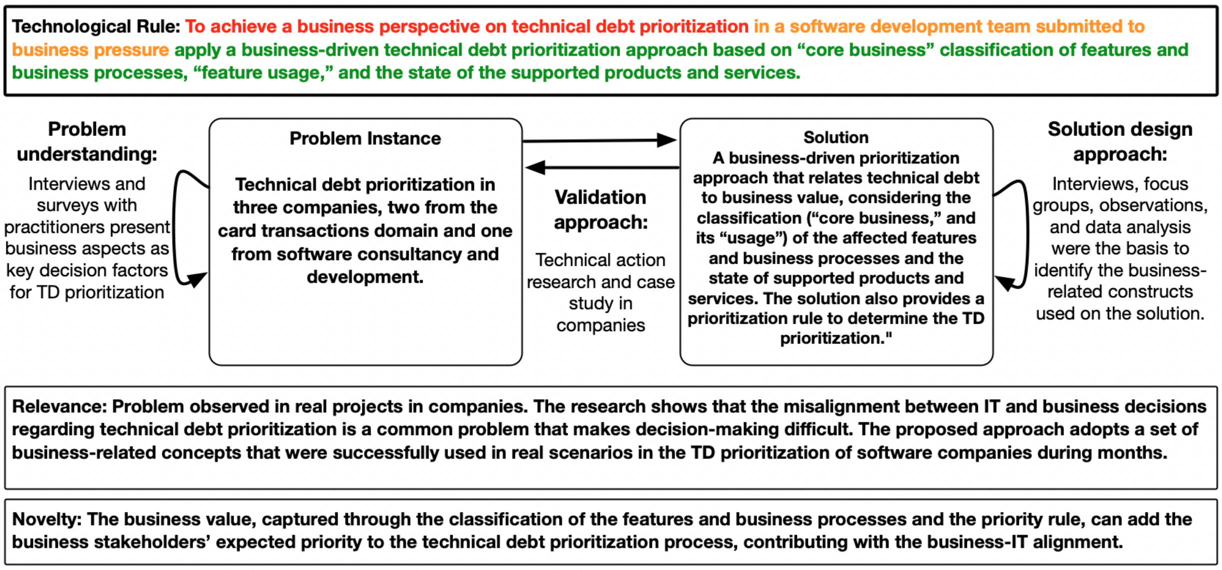


Figure 7: Design Science Template

Figure 7 presents the key takeaways of this DSR, using the visual abstract template proposed by [Storey et al. 2017]. The following technological rule expresses the scientific knowledge gained from this work:

To achieve *a business perspective on technical debt prioritization* **in** *a software development team submitted to business pressure* **apply** *a business-driven technical debt prioritization approach based on "core business" classification of features and business processes, "feature usage," and the state of the supported products and services.*

Problem understanding: after a set of interviews, and observations in routine decision-making, we understand that business aspects are key decision factors for TD prioritization.

Problem instance: "Technical debt prioritization in three companies, two from the card transactions domain and one from software consultancy and development."

Solution design approach: "Interviews, focus groups, observations, and data analysis were the basis to identify the business-related constructs used on the solution."

The solution: "A business-driven prioritization approach that relates technical debt to business value, considering the classification ("core business," and its "usage") of the affected features and business processes and the state of supported products and services. The solution also provides a prioritization rule to determine the TD prioritization."

Validation approach: "Interviews, focus groups, observations, and data analysis were the basis to identify the business-related constructs used on the solution."

Relevance: Problem observed in real projects in companies. The research shows that the misalignment between IT and business decisions regarding technical debt prioritization is a common problem that makes decision-making difficult. "The proposed approach adopts a set of business-related concepts that were successfully used in real scenarios in the TD prioritization of software companies during months.

Novelty: The business value, captured through the classification of the features and business processes and the priority rule, can add the business stakeholders' expected priority to the technical debt prioritization process, contributing with the business-IT alignment.

4.3 Business-driven Technical Debt Prioritization

Figure 8 describes the elements of the business-driven technical debt prioritization framework, called Tracy [Rebouças de Almeida, Treude e Kulesza 2019]. The approach is constructed around the "priority canvas," detailed in the next section. Besides a technical debt set to be prioritized ("technical debt list"), the Tracy

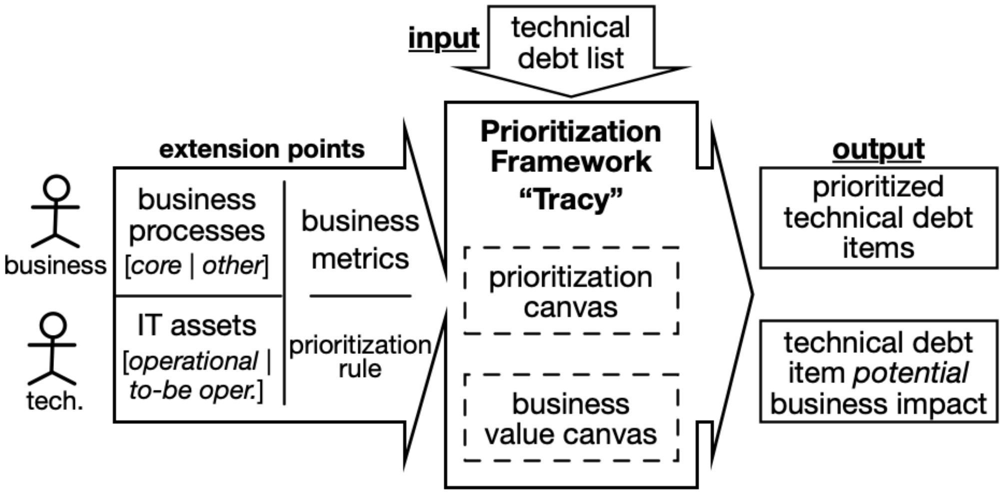


Figure 8: Components of the technical debt prioritization framework

framework uses configuration items and IT Assets to identify the affected business aspects. The value sources and business impact are the elements that represent the business value. The prioritization rule defines how the technical debt items must be prioritized. Finally, the priority canvas guides the output "prioritized technical debt items," and the business impact canvas guides the other output "technical debt business impact". All elements are detailed in the following.

4.3.1 Technical Debt List

The technical debt list can be registered directly in the TDM tool or imported from an existing issue tracker system.

4.3.1.1 Configuration Item

A term used by SWEBOK [Bourque, Fairley e Society 2014] and ITSM [UK 2012] to refer to a managed artifact. In our context, a configuration item (CI) is an artifact that can be affected by technical debt, such as a code fragment, a class, a library, a module, a system, a database, a server, an architecture element, or a service. Configuration items are generally from the technical stakeholders' domain whereas business stakeholders often do not master information about CIs and their responsibilities. A CI can be composed of and dependent on other CIs.

In our model, a configuration item can have one of three states:

- **operational (oper)**: an artifact that is in production, being directly or indirectly used by customers or users;
- **to-be operational (to-be)**: a new artifact that is under development or under planning;
- **legacy**: an artifact that is planned to be discontinued or replaced by another one.

A technical debt item affects one CI. For example, a "test debt item" can affect a "module of service"; a "build debt item" can affect an application; and an "architectural debt item" can involve server instances. Configuration items support IT Assets.

4.3.2 IT Asset

IT Asset is an abstract concept representing any solution, product, service, or mobile app that is part of the company portfolio. IT Assets are supported by configuration items. This concept's objective is to detach the company's solutions from the technical elements that implement them. Both technical and business stakeholders usually understand this concept. For example, at the business level, an e-commerce software system is evaluated independent of its technical implementation. It can even share configuration items with other IT assets, e.g., white-label solutions.

IT Assets have the same (but independent) states as configuration items. For example, a *to-be operational* IT asset can be composed of operational CIs (e.g., in the case of a new planned system that uses existing systems). On the other hand, an "operational" IT asset must be composed of at least one "operational" CI. For example, a new "to-be" e-commerce mobile application planned to be released in a few months can be composed of "to-be" CIs, e.g., iOS and Android mobile apps, and operational microservices.

4.3.3 Value Source

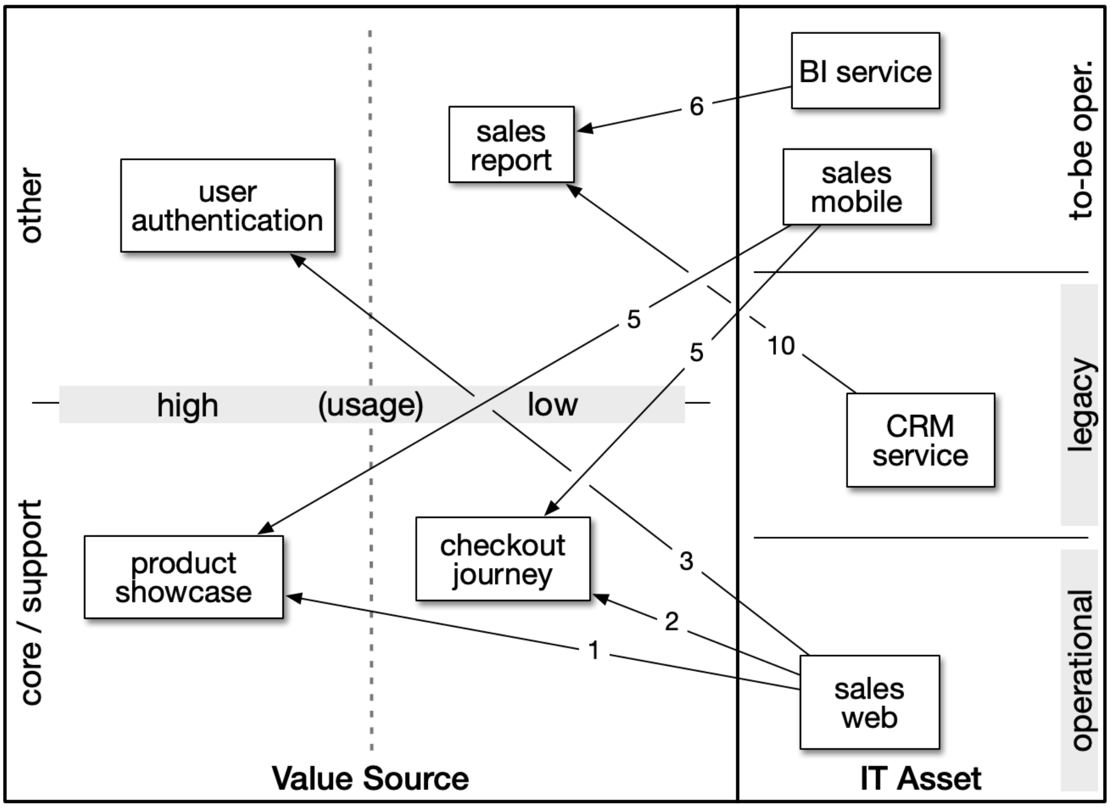
In the previous version of this approach [Rebouças de Almeida et al. 2018, Rebouças de Almeida, Treude e Kulesza 2019], we used the concept of "business process" [Rosing August-Wilhelm Scheer 2015] to identify the business value of a system. Although every system's features somehow affect a business process, the concept is not "common sense" among stakeholders. Both business and technical stakeholders have a different understanding of what a business process is. Sometimes, the business processes or activities are perceived as the "customer journey" from the marketing and UX perspective [Lemon e Verhoef 2016, Caddick e Cable 2011]. Besides that, some key features create value, for example, an executive report that supports decision making. Thus we called this abstract concept a Value Source. Value Sources are everything that creates business value from an IT Asset.

A value source can be classified as *core*, when it is part of the core business of a system, or *other*, when it is not. For example, in an e-commerce solution, the features supporting the customer's buying experience are the core business. On the other hand, the "management of past purchases" feature is not core business.

The value source can also be classified regarding its *usage frequency*. The *usage frequency* defines how frequently a value source is being used. This information can be obtained by monitoring tools or based on the stakeholders' perception.

Note that to help understanding, we will refer to the relationships between IT Assets or configuration items and value sources as follows: an *oper/core/high* IT Asset refers to an operational IT Asset related to a *core* value source that has *high* usage. Similarly, a *to-be/other/low* IT Asset refers to a

Figure 9: Priority canvas



not-yet operational IT Asset that affects a value source that is not core-business and has low usage frequency.

4.3.4 Priority Canvas

The Priority Canvas is a board used to visualize the main entities involved in technical debt prioritization. This board is used to help stakeholders visualize and discuss IT assets and value sources and their relationships. The board's objective is to guide the participants with exercises to think about IT Assets, value sources, and their classification. The participants can look at “the same page” and discuss business-value perceptions.

Figure 9 shows the board with an illustrative example where the *operational*,

Table 10: Priority Rule (example)

	Value Source		
Config. Item	**usage**	*core*	*other*
operational	*high*	1	3
	low	2	4
TO-BE operational		5	6
legacy	*high*	7	9
	low	8	10

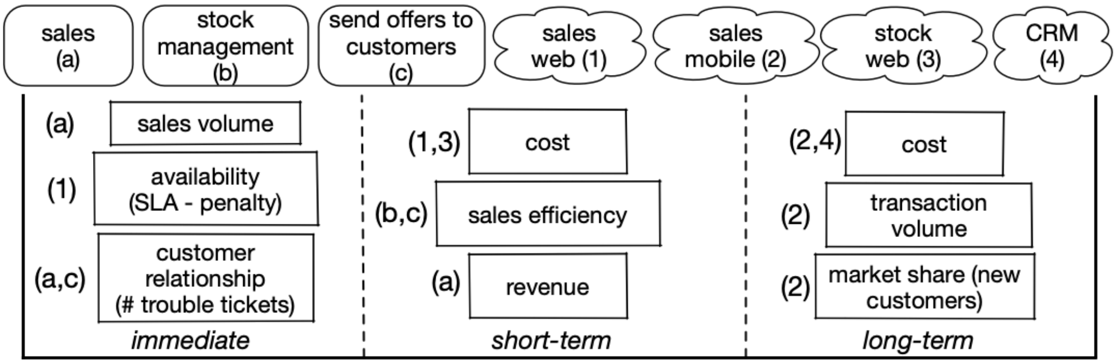
legacy, and *to-be operational* IT Assets support value sources classified according to their business value (*core/support* and *other*) and their usage frequency (*high* or *low*). The Value Source's *usage* attribute can only be related to operational and legacy IT Assets. Since a *to-be* IT Asset is not yet being used, we do not consider its usage frequency for the technical debt prioritization.

4.3.4.1 Priority rule

A priority rule is based on the relationship between IT Assets and value sources. A priority rule classifies the technical debt business priority from 1 (highest priority) to 10 (lowest priority), assigning the relationships between IT Assets and value sources. For example, if we assign a priority 1 to the *oper/core/high* relation, we consider that a technical debt item that affects a highly used core-business value source of an operational asset will have the highest priority. On the other hand, a priority 10 assigned to *legacy/other/low* will set the lowest priority to technical debt that affects non-core-business value sources with low usage supported by legacy IT assets.

Table 10 shows the example priorities used in the canvas (Figure 9). In the example, the relation between operational IT assets (sales web) and the highly used core value source (product showcase) has the highest priority 1. Since "sales mobile" is a *to-be operational* IT Asset, the priority between "sales mobile" and *high* and *low*-use *core* value sources are identical: 5.

Figure 10: Business impact canvas (example)



Note that the rules are flexible. It is possible to assign any scale of numbers and group different relations with the same priority. For example, in our study (Table 21), product owners P6 and P7 suggested grouping technical debt in priorities 2 and 3, respectively. In Section 5.5.2, we observe that many factors may affect the prioritization rule. In the study, we also mapped the priorities to the *high-medium-low* scale so we could compare them with the technical priority: priorities 1 to 3 were considered *high*, 4 to 6 *medium*, 7 to 9 *low*, and 10 *lowest*.

4.3.5 Business impact canvas

The *Business-impact canvas* (Fig. 10) is a complementary tool to improve the prioritization decision making with existing and planned business metrics relevant for technical debt management. It is the place where each value source and IT asset is related to business metrics. Each metric may have immediate, short-term or long-term business impact. This canvas is a tool to help stakeholders identify and classify the business value created by business processes and IT assets. The canvas aims at determining what is the potential immediate, short-term, and long-term business impact of technical debt which affects an IT asset. Depending on the company or project strategy, the time periods can be different from ‘immediate, short-term, and long-term’.

To identify the metrics, one must consider technical debt as a risk factor that may affect the business value [Allman 2012]. For each value source and IT asset, one must identify how they affect business, objectively. For example, in Figure 10, *a technical debt which affects* the *sales web* IT asset may have an immediate potential business impact on *customer A*, and on the *availability*. Additionally, the sales mobile IT asset is expected to impact customer A in the long-term.

4.4 Related work

Several secondary [Ampatzoglou et al. 2015, Ribeiro et al. 2016, Fernández-Sánchez et al. 2017, Lenarduzzi et al. 2019] and tertiary [Rios, Neto e Spínola 2018] studies analyze technical debt research. With regard to technical debt prioritization, it is a common finding that the criteria, tools, and approaches used to prioritize technical debt lack a business perspective. Lenarduzzi et al. [Lenarduzzi et al. 2019] conducted a systematic literature review on technical debt prioritization and identified only three papers that use business-related constraints. They highlight that based on most surveys conducted with practitioners, customer and business factors are the most important to consider when prioritizing technical debt. However, only a few papers addressed such factors.

Ribeiro et al. [Ribeiro et al. 2016] identified 14 decision-making criteria that can be used by development teams to prioritize the payment of technical debt items but only one of them considers the business aspect of cost-benefit analysis. Ramasubbu and Kemerer [Ramasubbu e Kemerer 2019] proposed a three-step normative process framework that incorporates steps for managing technical debt in commercial software development. The process is aligned to PMBOK practices and considers the cost of quality metrics and risk of financial loss as business impact. Different to our approach, they do not use a business process or a wider business-value perspective.

4.5 Conclusion

In this chapter, we have presented Tracy, a decision-making framework that prioritizes technical debt considering how IT assets support a company's business processes, thus providing a new perspective on technical debt management. Information about the potential business impact of each technical debt item is crucial to support decisions among stakeholders with different roles. Tracy was constructed using Design Science Research [Wieringa 2014], with the participation of 49 practitioners over six months.

5 Business-driven Technical Debt Prioritization: a Multiple Case Study

Earlier versions of the work in this chapter appears in the proceedings of the International Conference on Technical Debt - *best paper* 2021 [Rebouças de Almeida et al. 2021]

5.1 Introduction

In this chapter, we report the results of two case studies to evaluate the approach presented in the last chapter 4 to support the technical debt prioritization from a business perspective. It contributes to the alignment between technical and business perspectives for technical debt prioritization.

The first case study took place at Phoebus Technology,[1] a company that currently provides electronic payment, credit card processing, and sales processing solutions for more than 90 customers, including supermarket chains and credit card network stakeholders, e.g., credit card processors, banks, acquirers, and merchants. We will refer to it in this chapter simply as "Phoebus" or "case 1". The

[1] `http://www.phoebus.com.br/`, `https://www.paystore.com.br/en`

second case study took place on a startup developing an innovative social network solution (the startup is anonymous due to contract restrictions). We will refer to it on the text simply as "startup" or "case 2".

Business-driven technical debt prioritization involves information about IT artifacts, different stakeholders, their perspectives, and decision-making. To investigate how business decisions affect technical debt prioritization, we ran two industrial case studies to answer the following research questions:

RQ1: How does the proposed business-driven approach impact technical debt prioritization?

To answer this question, we applied the proposed approach supported by a tool in *case 1*, where we associated technical debt items to business-value elements; we identified and solved conflicting business perspectives among stakeholders; and, finally, we observed a higher downward trend in the amount of technical debt that has high business priority.

RQ2: What are the business stakeholders' perceptions regarding factors that influence technical debt prioritization?

By answering RQ2, we found that much goes on behind the scenes regarding prioritization decision making. After a set of interviews and focus groups, in *case 1*, we identified eight business factors that affected the stakeholders' decision-making.

RQ3: What are the benefits and potential improvements of running the approach in a real scenario?

After running the two case studies, we interviewed business and technical participants to collect their perceptions of running the business-driven TD prioritization in their routine. The results show that the teams use different approaches to manage technical debt identify and prioritize. The business perspective plays a relevant scenario in the decision-making.

5.2 Case studies

The case studies [Runeson et al. 2012] are part of the evaluation stage of a Design Science Research (DSR) [Wieringa 2014] project, where the proposed approach for business-driven technical debt prioritization was designed [Rebouças de Almeida, Treude e Kulesza 2019].

The three phases of the DSR were:

1. **The exploratory phase** - where we understand the constructs (concepts, relationships, rules, and motivation) related to technical debt decision making.
2. **The engineering phase** - where the first version of our approach is designed.
3. **The evaluation phase** - where the proposed approach have the first preliminary evaluation in a Technical Action Research [Wieringa 2014] in an industrial scenario.

This study extends the evaluation of the approach through two case studies that took fifteen-months (five plus 10 months), instrumented by a technical debt management (TDM) tool that allows us to handle real data and support real technical debt prioritization decision making.

To answer RQ3, after the end of both case studies, we ran a semi-structured interview with the product owner and the technical leader of the two teams (cases 1 and 2). The topics of the structured interview are below:

- Describe how the team manages technical debt.
- How does the team identify technical debt?
- How does the team prioritize technical debt?

- How does the team plan the technical debt payment?
- How often the team uses Tracy-TD?
- What are the suggestions to improve the technical debt management approach?
- What are the main benefits of using Tracy-TD to manage technical debt?

5.3 Case study 1

To answer the research questions, we ran the *case 1* during five months with a team from Phoebus Technology, a company that provides systems for the electronic payment market. The company uses an agile development process with the team's structure inspired by Spotify's Squad model [Kniberg e Ivarsson 2012].

The teams (or squads) are composed of multidisciplinary roles and are responsible for a set of products and services. Each squad has a product owner (PO), who works as a business analyst responsible for concerns such as customer contact, feature specification and prioritization, and delivery planning. The PO reports directly to the CEO. Consisting of developers, testers, software architects, and other roles, the squad is led by the Squad leader, a senior technical professional responsible for managing the squad's development routine and ceremonies, such as sprint planning and creation, and assignment and monitoring of development tasks.

5.3.1 Selected team

The squad in which the case study took place comprises the PO, the squad leader, six developers, two testers, and an architect. The team is responsible for maintaining eight products, services, and mobile applications that address the business of seven corporate customers. The systems comprise sales and other transaction solutions, business intelligence, integration solutions, and mobile applications that serve supermarket chains and sub-acquires.

The team uses Redmine[2] as the issue tracking system to maintain a backlog of requirements, development activities, bugs, and other activities related to the development process. They did not manage technical debt in a structured way before this case study. They are subjected to high business pressure, frequent changes in the prioritization of features, and creating new products and features to make sales presentations.

Unlike the other company's squads, the selected squad has a low dependency on other teams and other products maintained by the company. It is affected by constantly changing business priorities, since they are responsible for a company's new business area, attending to new customers. The product owner also has an active role in the specification of products and services features, prioritizing them, and planning the system's deliveries.

The company provided us with access to collect data and participate as an observer in several meetings and ceremonies. We as researchers also had permission to schedule meetings with all participants, and we were able to frequently access the top business level, like the CEO (Chief Enterprise Officer) and CSO (Chief Strategic Officer).

5.3.2 Case study 1: protocol

To answer RQ1 (How does the proposed business-driven approach impact the technical debt prioritization?), first, we collected data and prioritized technical debt (1st goal below); and after (Figure 11), we classified the value sources and defined a prioritization rule (2nd goal below) to run and evaluate the business-driven technical debt prioritization.

To answer RQ2 (What are the business stakeholders' perceptions regarding factors that should influence technical debt prioritization?), we ran a set of inter-

[2] `https://www.redmine.org/`

views and focus groups with business and technical stakeholders to discuss their perspectives about business value and the technical debt prioritization rule, during steps 2.1 and 2.2 (cf. Figure 11).

1st Goal: Relate the technical debt to its affected value sources to identify its business priority. Our objective was to trace the configuration items affected by a technical debt item to its impacted IT Assets and Value Sources. To achieve this goal, we collected (step 1.1) a list of *technical debt items*, (step 1.2) information about *configuration items* and their relationship, (step 1.3) *IT Assets*, and (step 1.4) *Value Sources.* After the initial data collection, we (step 1.5) *tested the technical debt prioritization* with a business stakeholder to have a first evaluation of the relationship between technical debt and value sources.

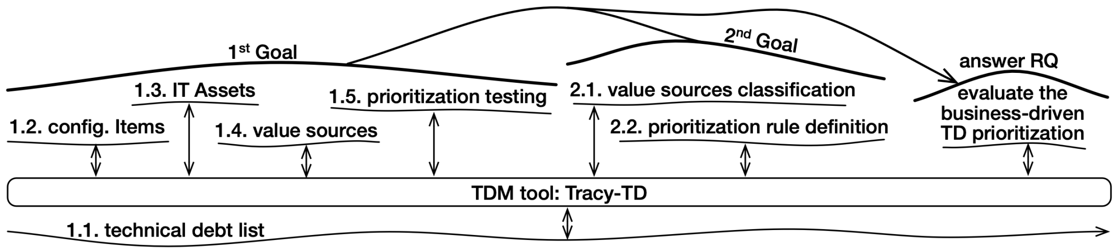
2nd Goal: Classify the business value of value sources and find a consensus regarding business value and technical debt prioritization rule. As we verified in chapter 3, different stakeholders have different opinions about the business perspective which makes finding a consensus on the value source business classification a necessity.

After achieving the first goal, we collected the (step 2.1) *value source business value classification* from different stakeholders as well as the different perspectives for the (step 2.2) *technical debt prioritization rule.* Our objective was to identify disagreements between stakeholders and promote a consensus about the value and prioritization criteria to be used.

Since the correctness of the data collection is essential to enable a correct technical debt prioritization, during all steps, at least two participants of the same profile (technical or business) reviewed all collected data. Also, participants could review and update all data regularly using the TDM tool. At the beginning of the case study, we established a research policy where the study participants were the sole responsible for providing and updating the data in the TDM tool.

On the first day, the whole team participated in a training about the main technical debt types and concepts, and the concepts of the proposed approach: the configuration items, IT Assets, value sources, and business impact. The training about technical debt became part of the team's onboarding protocol for new members and was repeated twice during the case study when new members became part of the team.

Figure 11: Case study 1 protocol



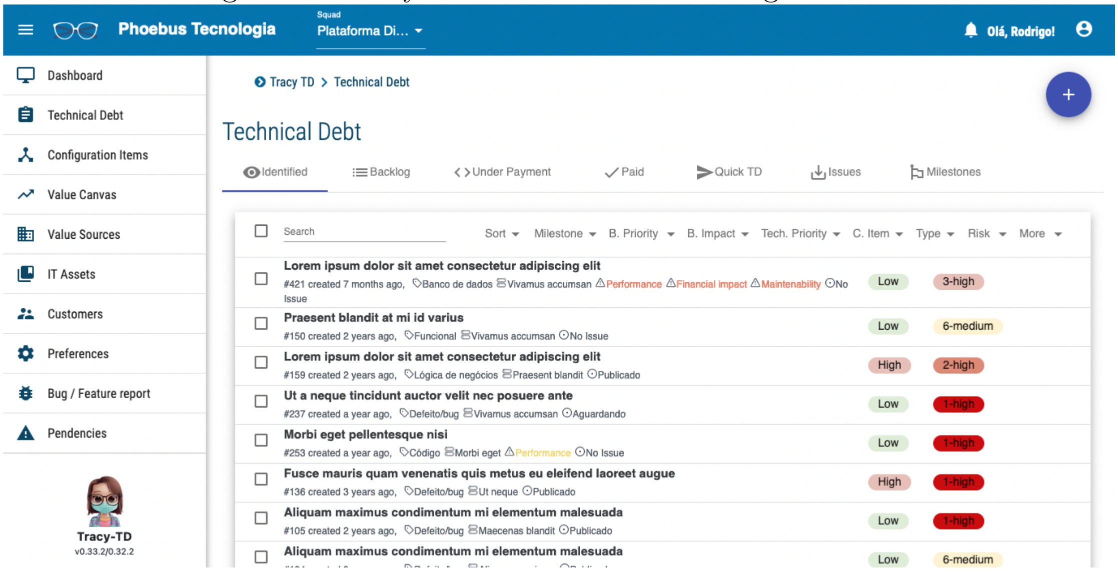
5.3.3 Tracy-TD: The TDM Tool

To integrate the research into the team's routine, we developed a tool (Figure 12) to collect data, evaluate the model, and support technical debt prioritization. The TDM tool enabled monitoring the team's issue tracking system and manage the technical debt lifecycle and prioritization. Thus, it was possible to conduct the technical debt management with the team's development activities without interfering with their routine. It was also possible to enable and disable the tool's functionalities to execute research activities with the participants.

The development of the tool started in September 2019. Since the model was likely to evolve to address new requirements, the TDM tool underwent parallel development with frequent releases during the case study. It had 11 major and 45 minor updates, with features, enhancements, and bug fixes during the case study (April 2020 to September 2020).

In the following, we detail each step of the process of data collection and analysis.

Figure 12: Tracy-TD technical debt management tool

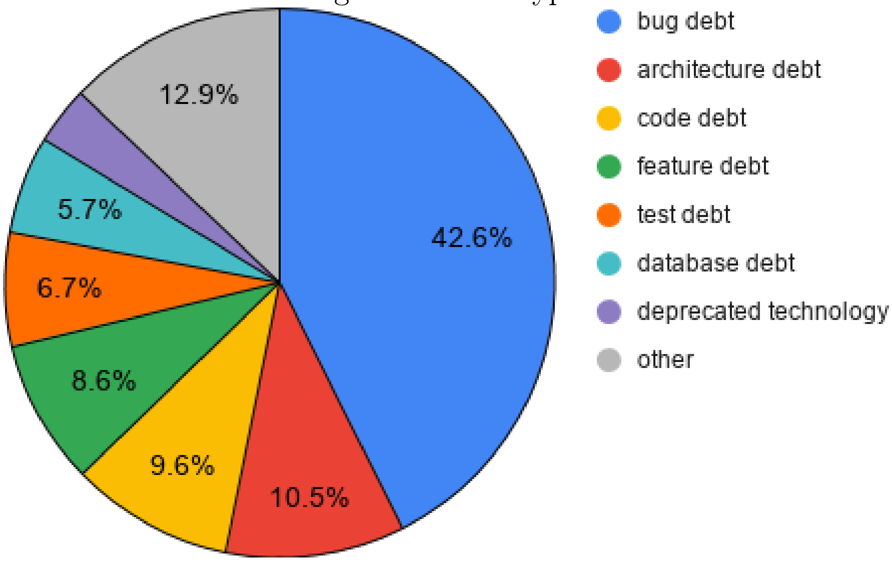


5.3.4 Technical debt list

Since the team did not use a structured approach to technical debt management, the initial set of technical debt items (Figure 11, step 1.1) was obtained through a focus group with all squad participants. The participants were asked to discuss all existing technical debt they were aware of in the systems they work with. All debt items identified in this focus group were stored in the TDM tool by the Squad Leader. The discussion and this exercise contributed to strengthening the concept of technical debt in the squad.

After that, the team started using the TDM tool to store all technical debt they identified during their work. Identification of technical debt can occur at any time, by any team member. Each technical debt item is registered using the following information: its name, description, creation date, payment date, type, technical priority, and technical effort. The technical priority is a priority given by a technical leader, used to prioritize the issues in the backlog. The technical effort (high-medium-low) is the evaluation of the effort necessary to pay the technical debt.

Figure 13: TD Types



Many technical debt items are identified during planning and problem-solving meetings. Once identified, it is possible to register the technical debt in the TDM tool and also to import any issue marked as technical debt from the issue tracking system. During the case study, 30% of the technical debt items were registered using the TDM tool, and the other 70% were imported from the backlog issues classified as technical debt.

In our case study, 209 technical debt items were reported and managed in the TDM tool. We started with a total of 140 technical debt items identified from the existing backlog and focus group. Ten of them were paid before the case study start date. During the case study, 69 new TD items were identified, mainly during the sprint execution and evaluation. In the end, a total of 62 TD items were paid, and 137 remained unpaid during the case study period (6 months). The most frequent type of debt was bug debt (42.6 %), followed by architectural debt (10.5%), code debt (9.6%), feature debt (8.6%), and test debt (6.7%) (Fig. 13). Bugs were considered technical debt if they were expected to be solved during the sprint period and were postponed for some reason [?, ?, ?].

5.3.5 Configuration Items

Since the team did not have updated documentation of its maintained configuration items, we started collecting data about configuration items (Figure 11, step 1.2) through interviews and discussions involving the squad leader and the architect. The analyzed data included architecture diagrams, source code structure, modules, services, infrastructure elements, and their related dependencies. All data were stored in the TDM tool, and the existing technical debt items were related to their affected configuration items.

When the participants registered new technical debt items, they could select existing CIs or create new ones if necessary. This process was repeated at each technical debt registration and contributed to the refinement of the configuration item granularity.

We considered the granularity of systems, services, and their main modules to describe the configuration items. A module can be a subset of a system which has a well-defined responsibility. For example, an e-commerce system (composed CI) is composed of authentication and payment modules. It can also be an instance of a system or application. For example, a white-label mobile application (composed CI) is composed of different customized systems of the same application. Note that composed CIs share the code with their components.

5.3.6 IT Assets

IT Assets are shared between business and technical domains, so the IT Assets were obtained by interviewing the product owner and the squad leader and were reviewed by the CEO (Figure 11, step 1.3). Since the configuration items are from the technical domain, the relationship between IT Assets and their supported configuration items was provided by the squad leader and reviewed by the architect. This case study involves eight IT Assets, with five of them in the *operational* state, two *to-be operational*, and one *legacy*.

Table 11: Configuration item types

Type	Quantity	%
module	21	34.4%
mobile application	17	27.9%
application	13	21.3%
service infrastructure	7	11.5%
storage service	2	3.3%
external system	1	1.6%
total	**61**	**100.0%**

5.3.7 Value Sources

The value sources are business domain entities. Therefore, we obtained the first set of value sources and their relationship with IT Assets (Figure 11, step 1.4) from the PO and squad leader, with a review by the CEO. After that, every technical debt item was linked to existing value sources and—following the same process of the previous entities—if the user missed a value source, it was possible to register a new one. Moreover, this set evolved during the usage of the TDM tool.

5.3.8 Prioritization testing

After the first value source set collection, we ran a preliminary technical debt prioritization (Figure 11, step 1.5) considering the first evaluation of the value sources provided by the PO. The objective was to test the technical debt prioritization with a controlled scenario, and verify whether we were missing something. We organized three sessions with the PO, where we asked him to classify the technical debt from a set of existing issues from their backlog. For the first set, we randomly selected old issues (20 out of 250 issues from January 1 to December 31, 2019), and in the second set, we selected newer issues (20 out of 123 issues from

January 1 to June 30, 2020). Twenty issues were an appropriate number to fit into a one-hour session. We selected old issues for the first session to avoid that his classification would be affected by current business pressures. In other words, if an issue is old, it was not prioritized for a long time and tended to have low priority. This would be a good way to understand the variables that are not addressed by our model. In contrast, in the second session, we selected recent issues to verify whether current business pressure would interfere with the classification. Finally, we ran a third session where we asked him to freely select issues that he thought have a high business priority and should be selected to be paid. He selected 13 issues. After the three sessions, we identified that we must include the "usage frequency" variable in the model to improve the value source evaluation. The CEO and the technical leader also confirmed and agreed with the new variable.

5.3.9 Value source classification

The penultimate step to enable the business-driven technical debt prioritization is the classification of the value sources (Figure 11, step 2.1). The stakeholders involved in the decision making must agree on the value source classification to avoid conflicts in technical debt prioritization.

To check the alignment between the participants about how they perceive the business value of their software systems, we asked five different stakeholders (PO, CEO, two developers, and one tester) to categorize 46 value sources as "core-business" or not. They also classified the value sources as "high" or "low" usage regarding their business value and usage frequency. We opted for binary classification to help the stakeholders decide and converge on the classification of what is sufficient for decision making.

During our discussions, we confirmed that the binary business-value classification makes sense. One of the participants, for example, said to convince others, "there is no medium core business, a feature is core or is not." However, the usage frequency can be improved to a range, and also receive the input from a moni-

Table 12: Priority rule considered in case 1

(Case 1)	V. Source	business value	
Config. Item	usage	*core*	*other*
operational	*high*	1	3
	low	2	6
to-be operational		5	7
legacy		8	10

toring system. The binary *usage frequency* classification must reflect the relevance of the usage of a value source for decision making. "High" means that the usage frequency is relevant from the business perspective, and "low" that it is not.

5.3.10 Priority rule definition

To understand the perspectives behind the business prioritization, we evaluated the prioritization rule with a set of five POs, the CEO, and the CSO (Figure 11, step 2.2). To avoid interference between participants, we ran individual interviews and asked them to provide a prioritization rule based on their context (e.g., products, squad, business forces). Table 21 shows the different prioritizations.

After the individual interviews, we conducted a focus group with all participants to discuss the different prioritization scenarios provided in the interviews. We first asked the participants who had different perspectives from the majority to discuss their proposal. For example, P6 prioritized value sources that have high usage frequency. After that, we opened the discussion. Since there is a hierarchy between POs, CEO and CSO, the POs talked first, to reduce bias in their opinion.

5.4 Case study 2

After running the first case study, we followed the adoption of the business-driven technical debt approach on a second case study to answer the research question RQ3, and analyze how the business-driven prioritization would affect the stakeholder's decision-making.

The company is a startup composed of one team. They use a development process based on Scrum; they have nine members with different roles: developer, DevOps engineer, team leader, and product owner. Their solution was not made public until the time of this publication.

The main aims of the second case study were to apply the approach in the context of a startup and collect feedback from the participants. All the companies involved in this research were big and medium companies until this case.

5.4.1 Case study 2: protocol

To answer RQ3 - *What are the benefits and potential improvements of running the approach in a real scenario?*, we run the following steps to deploy the approach on a new team. After that, we collected data and followed the team for ten months. The steps on case two and the differences from case one are shown in table 13.

First, we installed Tracy-TD, the technical debt management tool developed during the first case study. The tool handles the necessary data required to perform the business-driven TD prioritization. After that, we performed training about technical debt, as done in case 1.

The technical leader registered the first set of TD items using Tracy-TD. Next, we identified the first set of configuration items, as in case 1; after that, the PO, followed by the team leader review, registered the first set of IT Assets and Value Sources. Once the team was developing a single mobile app with an administration portal, these steps were straightforward. The PO classified the value sources as its

Table 13: Comparison between the two cases

Step	Case 1	Case 2
TDM Tool (Tracy-TD) adoption	The tool was developed during Case 1 (5.2.1.3)	The team used the tool previously developed
Training about technical debt	yes	yes
Collect the first list of technical debt items	Done by a focus group with all squad participants	Done by the team leader
Registration of TD items	using TracyTD	using TracyTD
Initial configuration Items identification	yes	yes
IT Assets identification	from the PO and squad leader, with a review by the CEO	PO defined and the team leader reviewed
Value Source identification	PO and squad leader, with a review by the CEO	PO, reviewed by the team leader
Value Source classification	First checked the alignment between five stakeholders, and then looked for a consensus	Classified by the product owner
Collect the value source business classification	from different stakeholders	with the team's PO
Prioritization testing	three sessions with the product owner	no
Technical debt prioritization rule definition	with different stakeholders	Team's PO and Team Leader
Understand the perspectives behind business prioritization	focus group with seven business stakeholders	no

business value (core business or not) and expected utilization (high or low).

We did not run a specific step to test the prioritization, as done in case 1, since the prioritization algorithm has been evaluated several times in the first case. Finally, the PO and Team Leader defined the prioritization rule in a joint session. In the first case, our objective was to understand the building blocks of different business stakeholders on defining technical debt prioritization. In the second case study, no other business stakeholders were available to perform a similar session as in case 1.

During the second case study, the team had the opportunity to mature the technical debt management process and create a culture on the team.

In the end of the case study, we ran a semi-structured interview with the product owner and the team leader to get their perceptions about their experience.

Table 14: Priority rule considered in case 2

(Case 2)	V. Source	business value	
Config. Item	usage	*core*	*other*
operational	*high*	1	3
	low	3	5
to-be operational		2	4
legacy	high	6	7
	low	8	10

5.4.2 Prioritization rule

The prioritization rule (Table 14) was defined after a meeting with the product owner and the team leader.

The rule has a change on the overall prioritization from case 1 Table 12. While Case 1 prioritized changes on operational assets (1-3 priorities), in case 2, we observe the same highest priority to technical debt that affects core/high, but the second priority (2) is defined to technical debt that affects to-be operational assets. As expected, since the team did not release their product yet. When asked why they defined the same priority (3) to other/high and core/low, the team leader said that until the meeting moment, they prefer to put technical debt that affects high-usage features and core/low on the same basket to force them to consider other criteria on the prioritization. "I do not want to have other/high technical debt listed before or after core/low ones. At this moment, It is not easy to decide which one is more relevant. It is a dilemma between core but with less usage and not core but highly used features." The legacy prioritization was the same as in case 1.

At the time of the priority rule definition, the team did not have operational assets, so they predicted the priority considered that the system was released. We highlighted that the priority rule could be changed when they change their mind.

5.4.3 Technical debt List

During the second case study, the team accumulated 139 technical debt items from 21 types (Table 16). The most frequent ones were code debt (34.5%), followed by test debt (20%) and architectural debt (6.5%). The other 39% are related to documentation, security, usability, and others. We can observe in Table 16 that the team put effort into paying most of the architecture debts (7 out of 9), documentation debts (6 out of 7), and the system's infrastructure debts (3 out of 4); they also paid debts of the following types: security, analysis, defect/bug, feature, service, configuration, events, and A/B tests.

The high payment of documentation debts took our attention. All the eight registered documentation debt from both cases have low technical priority. In this case, the documentation debt refers to the lack of external APIs documentation.

Figure 14 shows the technical debt accumulation during our second case study. Considering that the team did not release the application, they did not have operational or legacy assets. That is why all technical debt was classified with priorities 2-core/to-be and 4-core/other, so, for the sake of simplicity, we will refer to these priorities simply as 'core' and 'other'.

The technical debt with priority medium '4-other' had a slight variation - it started with 40 technical debt items and ended with 49. On the other hand, the team started with 8 TD items with high priority '2-core' and accumulated 56 items during the period, ending with 64 high-priority technical debt. We presented this fact to the PO, and he explained that they are focusing on core-business features during this period, so, since there are more core features, there must be more high-priority technical debt to be paid in the future.

From Table 16 we see that besides the accumulation of high-priority technical debt, the team put effort to pay 34 out of 85 (40%) TDs classified as 'core'. They did not put the same effort into paying technical debt with less business priority (15 out of 45 - 33%). In case 1 (Table 18), the team also paid more core business

Table 15: Case 2: TD Types

type	**backlog**	**paid**	**total**	**%**
Code	37	11	**48**	34.5%
Test: Automated test	26	2	**28**	20.1%
Architecture	2	7	**9**	6.5%
Documentation	1	6	**7**	5.0%
Business logic	3	2	**5**	3.6%
Security	0	5	**5**	3.6%
Usability	3	2	**5**	3.6%
Analysis	0	4	**4**	2.9%
System's Infrastructure	1	3	**4**	2.9%
Database	1	2	**3**	2.2%
Defect / bug	0	3	**3**	2.2%
User expectation	2	1	**3**	2.2%
Build	4	0	**4**	2.9%
Feature	0	2	**2**	1.4%
Dev. Infrastructure	1	1	**2**	1.4%
Service	0	2	**2**	1.4%
Communication	1	0	**1**	0.7%
Configuration	0	1	**1**	0.7%
Events / future messages	0	1	**1**	0.7%
Deprecated technology	1	0	**1**	0.7%
Production tests	0	1	**1**	0.7%
total	**83**	**56**	**139**	**100.0%**

Figure 14: Case 2: Technical debt accumulation

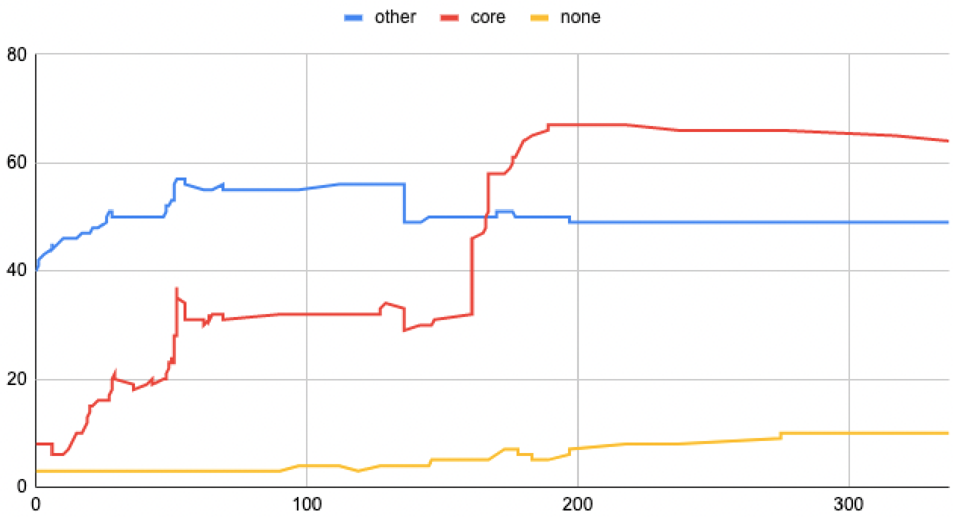


Table 16: Case 2: total of technical debt per business priority

Business priority	**backlog**	**paid**	**total**	**%**
2-core	51	34	85	61.2%
4-other	30	15	45	32.4%
none	2	7	9	6.5%
total	**83**	**56**	**139**	**100.0%**

and high-usage TDs.

It is essential to highlight that the technical debt items selected to be paid in both cases are not causing incidents. In other words, they were selected through technical debt management.

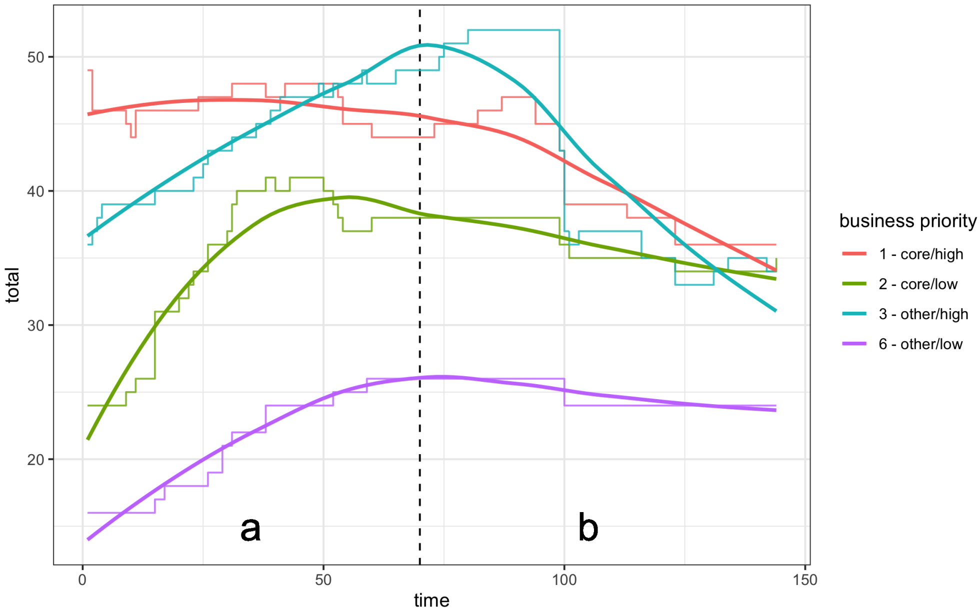
Table 17: Technical versus business priorities

	technical priority				
business priority	**high**	**medium**	**low**	**none**	**total**
1-core/high	21	12	25	-	**58**
2-core/low	3	8	32	1	**44**
3-other/high	16	12	31	2	**61**
6-other/low	2	3	22	-	**27**
10-other/legacy	2	-	-	-	**2**
not prioritized	2	-	2	3	**7**
Total	**46**	**35**	**112**	**6**	**199**

Table 18: % of technical debt payment

business priority	**previous (A)**	**identified (B)**	**total (A+B)**	**paid (C)**	**% paid (C) / (A+B)**
1-core/high	47	11	58	22	38%
2-core/low	24	20	44	9	20%
3-other/high	36	25	61	27	44%
6-other/low	16	11	27	3	11%
10-other/legacy	2	0	2	0	0%
not prioritized	5	2	7	1	14%
Total	130	69	199	62	-

Figure 15: Technical debt accumulation trends



5.5 Impact of the Business-Driven Approach

5.5.1 Answering RQ1: How does the proposed business-driven approach impact technical debt prioritization?

Technical versus business-driven technical debt prioritization: Table 17 shows the tension between the business and technical prioritization. From the 58 items with the highest business classification 1-core/high, only 21 (36%) also have a high technical priority and would be aligned with business priorities. The discrepancy is higher in the 2-core/low items category since 73% of the items have the lowest technical priority but may affect core features or business processes. **The difference between technical and business priorities shows that business prioritization is an additional dimension to support decision-making.**

We also analyzed the accumulated series of technical debt during the first case study period (143 days). Figure 15 presents the graph of the amount of technical debt classified according to its business priority. Each thin line represents the total number of technical debt items identified and paid in a day. For example, if two items are identified, and one is paid, the day has a total of one debt item added to the previous day. The bold lines are the technical debt item accumulation trends. It is important to highlight that we consider the item's identification date, not the registry date. The vertical dotted line indicates when we achieved goals 1 and 2 and started to use our approach to perform business-driven prioritization. The x-axis is divided into periods *a* (technical prioritization) and *b* (business-driven prioritization).

Table 18 shows the number of technical debt items registered before the case study, and the number of identified and paid ones during the study. Table 19 shows the percentage of paid items regarding their technical effort for the most paid items (1, 2, and 3). Our objective is to compare the trends during the period "b" and evaluate the forces behind selecting which item should be paid. The amount of paid technical debt classified as *1-core/high* (38%) and *3-other/high* (44%) was higher than the debt classified as *2-core/low* (20%) and *6-other/low* (11%). When we consider the paid technical debt's technical effort, we observe that the highest effort was dedicated to the items with the highest business priority (cf. Table 19). 22.7% of the highest business priority items had a high technical effort, while only 11.5% of the ones classified with business priority 3 had high technical effort. Among the paid items with the highest priority, 5 (22%) of them were of the "architectural debt" type, and no architectural debt with business priorities 2 or 3 was paid. **Despite the number of paid items with business priority 3-other/high being higher than the ones classified as 1-core/high, the team dedicated more technical effort to pay the technical debt with business priority 1-core/high.**

Table 19: Technical effort of the paid technical debt

	Technical Effort		
business impact	**high**	**medium**	**low**
1-core/high	22.7%	4.5%	72.7%
2-core/low	0.0%	25.0%	75.0%
3-other/high	11.5%	19.2%	69.2%

Table 20: Agreement on value source classification

	free-marginal kappa	
participants	business value	usage frequency
All	0.29	0.34
PO x CEO	0.29	0.78
PO x Squad Leader	0.47	0.51

5.5.2 Answering RQ2: What are the business stakeholder's perceptions regarding factors that should influence technical debt prioritization?

During the meetings to discuss the value source classification and the prioritization rule, the different perspectives were frequently apparent. It was possible to identify some factors that are behind what the participants considered in their arguments.

Below we discuss the scenarios of conflicts during the meetings of the value source classification and the priority rule definition. It is important to highlight that while we bring people from different projects together, it is expected that they have different opinions about the topic under analysis.

Conflict analysis of the value source classification: Table 20 shows the Kappa [Brennan e Prediger 1981] agreement among the participants, regarding their opinion about the business-value and usage frequency of the value sources. First, with all participants (45 cases, two categories, and five raters), we had 0.29

Kappa agreement on the business value and 0.34 Kappa agreement on the usage frequency. This shows how the perception of what is core-business or not is different among the team members. We observed the same level of agreement (0.29) on the business value and 0.78 Kappa agreement regarding the usage frequency between the business stakeholders (PO and CEO).

The disagreement between the CEO and PO was a result of a different evaluation dimension about core business from other study participants. The CEO's perception about the business value is customer and marketing-centric whereas the PO has a more functional-centric perception. For example, the CEO evaluated a report as core-business while the PO did not—for the CEO, the report "is on my sales presentations" but the PO considered that the report was a side-feature and was not related to what the product was supposed to do as core-business. The PO's classification was more aligned with the definition of a core business feature, but the report had enough business value to maintain its classification as a core business. Both agreed that the report had low usage.

Finally, the PO and Squad leader had a higher agreement for the two variables—0.47 on the business value classification and 0.51 on the usage frequency. Unlike the PO-CEO case, the perception of business value was aligned between the PO and the squad leader. Both share a routine of feature-driven decision-making, making them share the same level of perception of how core-business features and processes are. Their disagreement was mostly related to a lack of understanding of how features work and how processes affect customer routines and outdated information about business contracts and feature usage.

Incident versus Technical debt prioritization: Both technical and business stakeholders often evaluated a value source or technical debt priority considering an incident scenario. Many times it was necessary to highlight that technical debt and incident prioritization are different. For example, the authentication feature of all products and services was, at a first evaluation, classified as "core-business" because "if the authentication is not working, the user cannot use the system."

A way to expose the core value sources was to ask, "What are the customers paying for?". For example, in an e-commerce system, the customer does not pay for the system to authenticate users. Although essential for the system usage, the customer pays for selling the product, a core-business feature. On the other hand, if the authentication causes service disruption, the incident must be prioritized and probably will gain a high priority. Note that technical debt can cause incidents, and maybe debt should be prioritized if that is the case, but as a result of a different decision-making process.

The "authentication" feature example came up in one of our meetings. One of the participants realized that, besides proposing the authentication as a core-business, they did not prioritize an architectural debt item on their authentication solution for almost one year. Other technical debt items always gained priority and were paid before the authentication one. They said, "We still can wait for a while to pay it."

Another way to identify the core business features and processes was discussed by one of the participants. While convincing others that a feature was core-business, he asked, "Does the feature have any business rule on it? It is only a CRUD!". It was a good point to separate features that are only data management (for example, CRUDs for some entities) from other features that aggregate value and have more business rules, like features that process sales transactions.

The perception of risk affected the technical debt prioritization rule: Table 21 presents the different prioritization rules suggested by the participants after the focus group (Section 5.3.10). As we can see, there are two unanimous prioritizations: all participants consider that technical debt, which affects operational IT assets and core/high value sources, must receive the highest priority, while the ones that affect legacy IT Assets must receive the lowest priorities. More than one participant declared that *"Technical debt which affects operational core-business commonly used features and processes must be prioritized."*.

We also analyzed which categories received the highest classifications (1 to 3)

Table 21: Prioritization rules proposed by business stakeholders

CI state	value	usage	p1	p2	p3	p4	p5	p6	p7
oper	core	high	1	1	1	1	1	1	1
		low	2	2	2	2	3	4	3
	other	hgh	3	3	3	4	6	2	5
		low	6	7	6	5	7	5	7
to-be	core	-	4	4	5	3	2	2	3
	other	-	8	8	7	7	5	3	6
legacy	core	-	9	9	8	8	8	7	4
	other	-	10	10	10	10	10	9	10

(Table 22). High usage was the most prioritized, with 78.6%, followed by 60.7% of both core and operational. To-be received 64.3% medium and 50.0% of low prioritizations. Legacy was the lowest priority, with 92.0% of lower classifications, followed by to-be and other with 50.0%. The *high usage* prevalence is aligned with existing findings [Marciuska, Gencel e Abrahamsson 2013] that "in most of the cases the higher the usage, the higher the perceived value of a feature".

We refer to our participants using the identifiers P1 to P7. Participants P1, P2, and P3 provided the same prioritization with different perceptions about what is *high*, *medium* or *low*. For example, P2 considers *other/low* as a *low* priority (7) while P1 considers it a *medium* priority (6). They prioritize *core/high* while *other/legacy* had the lowest priority.

P5 was the only one who prioritized *core-legacy*. His context involves many solutions that must be certified, and the process of feature and system replacement has a delay due to certification procedures that involve third-party certification companies. He said: *"Since we cannot release features as frequent as the other squads, we have to prioritize technical debt on to-be configuration items. We cannot forget the existing debt of the systems and modules that will die (legacy) since they are slow to die."*

The perception of risk of a technical debt item attracted different perspectives.

Table 22: Percentage of the decomposed variables considered in the proposed technical debt prioritization rules

	high (1,2,3)	**medium (4,5,6)**	**low (7-10)**
core	**60.7%**	17.9%	21.4%
other	17.9%	32.1%	**50.0%**
high	**78.6%**	21.4%	0.0%
low	42.9%	35.7%	21.4%
oper	**60.7%**	28.6%	10.7%
to-be	35.7%	**64.3%**	**50.0%**
legacy	0.0%	7.1%	**92.9%**

For example, participants P1, P2 and P3 classified the priority of *core/to-be* as *medium* (4 and 5) while the others (P4 to P7) prioritized it as *high* (2 and 3). When asked to explain their motivations, P1 said that *"since the new feature or system was not delivered, we have more opportunity to handle the technical debt, negotiate and delay it, different from the case where the debt is already creating risk on an operational feature or system."*

On the other hand, participant P5 argued that *"the risk of debt on an operational feature is already known, we already decided before it went on-line. We know how many users can be affected by it and if it is causing incidents. On the other hand, the debt on a to-be feature is unknown. Will the user face incidents due to the debt? Will we have time and the ability to pay it in the future?"* Another argument in the direction of prioritizing technical debt that affects a *to-be* CI came from the perception that *"every debt we pay on a feature or system that is not yet operational (to-be), will be on an operational system in the future."* If the teams do not prioritize debt on to-be features and systems, they will have to prioritize it in the future, when it becomes operational.

After analyzing the participants' arguments, we identified eight fac-

Table 23: Factors that affected the TD prioritization,from the business perspective

factor	comment
Core-business value sources	Core value sources received the highest priority classification.
Usage frequency	Frequent usage received the highest priority classification.
Risk perception	TDs tended to have their prioritization aligned with risk perception.
Time negotiation flexibility	TDs in to-be CIs may have higher negotiation eligibility to pay the debts and may receive lower priority than the operational ones.
Incidents	Technical debt, which creates incidents tend to be prioritized.
Delivery frequency	A product or service with a low-frequency delivery routine may prioritize TDs that affect operational CIs.
Embedded systems targets for regulation and certification	Systems that depend on third party regulators, may prioritize technical debt that affects to-be and legacy IT assets and configuration items.
Customer and marketing-centric versus functional-centric perception	The business-perspective related to marketing, sales, and customer may conflict with a feature-related perspective.

tors that influenced the technical prioritization, considering the business perspective. Table 23 summarizes them. In this case study, it was unanimous that if a technical debt item affects a *core-business* feature or a feature that is heavily used by users (*usage frequency*), it should have a high priority. The risk perception related to the unknown impact of technical debt also affected the prioritization. A high-risk perception implied a higher TD prioritization.

The *flexibility to negotiate the time* to delivery also affected decision making. Situations where there is flexibility to negotiate and expand the delivery time tended to lower the technical debt priority. As one participant said: “If I have to choose between paying a TD in operation and one still in the development cycle, I choose the operation one. It is feasible to negotiate scope and time under development; in production, I cannot. The debt is already creating interest.” Participants also agreed that if a TD item is creating *incidents*, then its payment must be prioritized. The *frequency in which teams deliver* releases in production also affected the prioritization decision making. It is related to the “time negotiation flexibility” factor. Teams with low-frequency delivery have more time to deal with TD in the development cycle and prioritize TD in production (operational). The PO of a team that deals with *certified embedded systems* explained that the cost to change the system to pay a TD item in production is so high that they must prioritize all known TD in the development phase before being delivered. They also deal with many devices using legacy systems, which makes them prioritize technical debt in legacy systems.

Finally, we observed misalignment between the arguments that drive technical debt prioritization, according to different business-level participants. Product owners tend to have a more *feature-related perspective*, their motivations to define the prioritization were more related to deadlines, milestones, and scope. On the other hand, the directors and account managers tend to have arguments related to *marketing and sales perspectives*. Complementary perspectives should be aligned to support the overall technical debt management decision making.

5.5.3 Answering RQ3: What are the benefits and potential improvements of running the approach in a real scenario?

We observed that the business-driven technical debt approach regularly supported the prioritization decision-making (during sprint plannings) on both case studies. As reported by participants during the final interview, the prioritization approach also contributed to the communication and alignment between technical and business stakeholders. The concepts and data used in the approach have been shown to be adequate and feasible to collect and maintain in both cases (in both cases, the data collection was straightforward to collect and maintain).

The concepts “configuration item”, “IT Asset” and “value source” added an unnecessary complexity during the adoption of the approach. Both teams gave feedback that the concepts should be reviewed.

After interviewing the four professionals (two product owners and two team leaders) from the two case studies, we grouped their perceptions about three technical debt activities identified in their responses (identification, prioritization, and payment), and the TD management culture. Most of the findings are common to both teams from the two cases. When there is a particular aspect from a team, we will refer them as team 1 and team 2 (from cases 1 and 2, respectively).

Although debt identification is not directly linked to the prioritization approach, once the two teams started managing technical debts from our approach adoption, they reported their experiences with this activity, and we shared their perceptions above.

5.5.3.1 Technical debt identification:

Team 1 begins identifying technical debt during the requirements elicitation and analysis before the development effort. *"Sometimes some required feature demands an architectural solution that the team cannot implement on the expected* *delivery time. At this point, we need to choose between meeting the business expectations and implementing the optimal solution. The business expectations usually win the decision."*

Team 2 usually identifies technical debt during the daily meetings and "planning poker" sessions. In many cases, the technical debt arises in the declaration of impediments. As mentioned, *"some impediments lead to the necessity of some shortcut solutions to achieve the sprint plan"*. All the identified technical debt is registered, evaluated, and authorized by the team leader and the product owner on both teams.

Another moment where the technical debt is identified is during the code review on both teams. Team 2 mentioned that *"It is common to review code that does not follow internal patterns or pre-defined architecture constraints, or code that can be optimized. These are example situations where we can decide to accept the code and pay the debt in the future."*

Team 1 also identifies new TD items when high-priority bugs are discovered and demand a high effort. *"The bug is discussed with the squad leader and the PO to decide if it will be handled in the current sprint (and then create pressure for a TD on the planned tasks)."*

Team 1 is using Tracy-TD's code comment feature. This feature enables the developer to comment on a file and identify a technical debt. They mentioned this feature as a source for technical debt identification. The developer indicates the technical debt in the code or other artifacts (e.g., shell scripts, build config files, SQL files), and it starts to be tracked by the tool.

Finally, the status report or sprint review meeting is another moment where both teams identify technical debt. When developers discuss the problems and overcome difficulties, technical debt is highlighted by the team and registered on the TD management tool. Team 1's team leader also reports that it is frequent to identify technical debt in the existing code while refactoring the software to receive new features.

Team 2 is focused on finishing the features to release their product, so, as mentioned, they *"are in the technical debt accumulation phase."* Despite the accumulation, as reported in case 1, they also analyze the new feature's scope and prioritize technical debt present on the same artifacts that will be affected by the development intervention.

5.5.3.2 Technical debt prioritization:

Both teams used Tracy-TD as the technical debt management tool to prioritize technical debt. Team 1 said *"We use the tool to list and prioritize the technical debt, and since the tool is integrated with our task management tool, we can see how many technical debt items are created and paid at each delivery. We also prioritize the TD items, considering the rule that we defined."*

There is a discussion between the technical leader and the PO on both teams to prioritize the technical debt. The team leader from Team 1 mentioned: *"We discuss them [the TDs] with the PO. Sometimes they affect the customer and core functionalities, and the PO prioritizes them. Sometimes we identify that the technical debt will affect the system performance, so we also prioritize it."* The PO from team 2 argued that the business priority is the driver to the prioritization: *"It depends on the business priority. Debt that may cause a problem in production gains high priority. Core-business debt also gains high priority."*

5.5.3.3 Technical debt payment:

Both teams prioritize technical debt payment during the sprint planning, with the presence of the product owner. At this moment, they evaluate the effort to pay the debt item and its business priority. They use at least two approaches to manage the technical debt payment.

Feature versus technical debt: this is the scenario where technical debt has a high payment effort and a high business impact. In this case, the technical debt competes against prioritized features during the sprint planning.

Technical debt payment together with feature development: in this scenario, features are prioritized, and they check if some technical debt exists in the code that will receive the new feature, an enhancement, or a bug solution. The objective is to take advantage of the programmer's development context to reduce the debt payment's effort. In this case, the technical debt with small effort are candidates to be paid. This criterion was also previously observed by [Ribeiro et al. 2017] (localization of TD) and considered on their proposed multi-decision criteria approach.

5.5.3.4 Technical debt "culture":

The team leader from team 2 reports that the team reduced the frequency of technical debt identification, but in fact, they started deciding better which technical debt must be admitted. *"Today, the identification is less frequent than before."* Nowadays, we try to *"kill the debt before it is born. When it is necessary to create a debt, we discuss the problem with the product owner, review the requirement, and change some priorities to avoid debt creation."* He said that technical debt identification was most frequent during the first moments of technical debt management. When the team started tracking them and absorbed the culture of technical debt management, they started to avoid the creation of consciousness technical debt. *"During the last month, bugs were more frequent than technical debt identification."*

The members are open to declaring, identifying, and asking for technical debt creation on both teams. The team leader on team 2 says that *"Technical debt is expected every time. We are developing a new product under pressure, so at all moments, when a TD candidate appears, we must decide if we pay immediately or leave it to the future"*.

5.5.3.5 Benefits of using the approach to manage technical debt

Both teams declared that it is good to have a specific tool to track the technical debt and list and prioritize them to understand whether or not the technical debt affects the core business features. Both teams also highlighted the alignment between the team leader and product owner. The team leader from team 1 said that "It is good to talk with the PO and plan on the same page. For example, when we select a critical technical debt that affects a core business module, the product owner also understands that the debt item is critical." The PO from team 1 reported that "it is good to have the technical debt items together on the same system for easy visualization and prioritization."

5.5.3.6 How the approach can evolve

Both the PO and team leader from team 1 highlighted the necessity to increase the discussion about technical debt with higher-level management. They suggest adding technical debt to meetings with directors and sales areas. "There are opportunities to increase the interaction between teams about technical debt on shared systems, services, and infrastructure." They also suggest increasing the communication between different teams about technical debt management practices.

The PO from team 2 asks for a dashboard to show the percentage of the modules affected by critical technical debt. "A visualization of the systems and modules and how they are affected by technical debt. This way, we highlight the prioritization to deal with these modules."

The technical leader from team 2 suggested reviewing the concepts used by the approach. The term "configuration item" overlaps with other processes, like in project management. The terms "IT Asset" and "Value Source" are not intuitive.

5.6 Applying the approach

To apply the approach for business-driven technical debt management, we suggest to start with one team and ensure the participation of at least one technical stakeholder and one business stakeholder. An initial training with the team to reach common ground on what is technical debt builds the foundation for technical debt management and can be followed by a workshop with a technical leader and a business stakeholder to identify the initial set of IT Assets that will be the scope of the technical debt management. For each IT Asset, the initial set of value sources need to be identified and classified according to their business value (core/other) and usage frequency (high/low). To identify the initial set of technical debt items to be managed and to define the first prioritization rule, a workshop with the technical team is suitable. An iterative and incremental process should then be followed:

1. for each technical debt item, identify and register the affected configuration item. We suggest beginning with a two-level granularity, e.g., system/service and module;
2. relate the configuration item with one of the IT Assets;
3. relate the technical debt with one of the selected IT asset's value sources;
4. review the configuration items, IT Assets and value sources;

5.7 Threats to Validity

Internal validity: We had to avoid hierarchy bias during the study since we were dealing with conflict situations between different knowledge levels, e.g., business versus technical stakeholders, and different hierarchy levels, e.g., directors and product owners. To overcome this bias, we used an approach to collect information individually before opening it to discussion and letting participants at lower levels of the hierarchy speak first. During the last interview, we invited an external researcher to conduct the activity to avoid bias since we asked for feedback on the approach.

External validity: The presented results are related to two companies and two teams. They cannot be considered generalizable. However, the evaluated approach's building blocks do not use concepts or rules particular to the companies. The business factors that affect the TD prioritization resulted from the seven business participants' perceptions and we cannot claim generalizability to other participants.

5.8 Related Work

The research field currently lacks business-related criteria for decision-making and approaches for technical debt prioritization. A systematic mapping study [Ribeiro et al. 2016] identified 14 decision-making criteria that development teams can use to prioritize the payment of TD items. The identified studies concentrated on two types of debt (defect and design), and the only studies that consider business-related criteria are concentrated on "cost-benefit". Our work is positioned to contribute to filling this gap.

Ribeiro et al. [Ribeiro et al. 2017] present a strategy for TD management that uses multiple decision criteria to decide when to pay debt items off. Their work proposes a configurable multi-criteria decision approach based on weights assigned to 14 categories. Some of the criteria can be driven by business forces, like the customer, severity, and cost-benefit, but the approach considers the classification for each technical debt item individually, done by a software engineer. Their approach is different from ours since they do not consider the IT artifacts affected by the technical debt or its business value. We also work with the definition of a general prioritization rule applied to technical debt, despite the individual technical debt evaluation.

Table 24: Studies compared by [Alfayez et al. 2020]

Code	Study
S1	[Schmid 2013]
S2	[Guo e Seaman 2011]
S3	[Ribeiro et al. 2017]
S4	[Rebouças de Almeida et al. 2018]
S5	[Sae-Lim, Hayashi e Saeki 2016]
S6	[Snipes et al. 2012]
S7	[Plösch et al. 2018]
S8	[Guo, Spínola e Seaman 2016]
S9	[Aldaeej e Seaman 2018]
S10	[Vidal et al. 2015]
S11	[Letouzey e Ilkiewicz 2012]
S12	[Choudhary e Singh 2016]
S13	[Mensah et al. 2018]
S14	[Tornhill 2018]
S15	[Zazworka, Seaman e Shull 2011]
S15	[Al-Barak e Bahsoon 2018]
S16	[Codabux e Williams 2016]
S17	[Akbarinasaji 2015]
S18	[Fontana et al. 2015]
S19	[Harun e Lichter 2015]
S20	[Abad, Conference e 2015]
S21	[Seaman et al. 2012]

Our work is aligned with Martini and Bosch [Martini e Bosch 2015] who provide nine prioritization aspects for architectural debt, identified by business and technical participants. Their identified aspects (e.g., competitive advantage, specific customer value, market attractiveness) can be used to guide the classification of value sources. They also identified different conflicts regarding the prioritization between the POs and the software architects.

Table 25: Comparison of the studies limitations [Alfayez et al. 2020]

	Limitations						
Study:	Applicability	Communication among stakeholders	Computational complexity	Error proneness	Loss of information	Rank updates	Scalability
Tracy-TD (current)			(needs evaluation)				(hundreds of TD items and dozens of configuration artifacts and IT Assets)
S4		x	x				
S18, S23.1			x			x	x
S1, S6, S9, S16, S21				x			
S2, S17			x	x			
S14, S15, S18				x			x
S10, S22, S23.2	x			x			
S5, S7, S8, S13, S20					x	x	
S19				x		x	x
S12				x			
S3, S11				x			x

5.8.1 Comparing studies limitations

A recent systematic literature review on technical debt prioritization [Alfayez et al. 2020] reveals the scarcity of approaches that account for cost, value, and resource constraints as well as a lack of industry evaluation. They compare the limitations of 23 technical debt prioritization approaches, including our first multiple case study [Rebouças de Almeida et al. 2018]. We now compare the current state of our approach with the same studies, using the same limitation criteria. The studies are mapped in Table 24, and the comparison of the studies' limitations is shown in Table 25. Below we describe briefly how our approach overcomes these limitations.

Applicability: our approach is applicable in real scenarios since it was built on top of a few core concepts and applied in different industrial cases.

Communication among stakeholders: the current approach does not need the input from different stakeholders for each TD item. The necessary information is regarding the business value classification.

Computational complexity: *(Must be evaluated)*. Our current implementation handles hundreds of technical debt items with dozens of configuration items, IT Assets, and Value Sources with no performance issues. Each TD item is classified only once registered and updated when there is a change on the configuration items and IT Assets relationship graph. The most complex processing is the prioritization rule update, which triggers the update on every non-paid TD item. Considering that the rule update is not frequent, an eventual performance issue is irrelevant.

Error proneness: The approach uses a small set of information to classify the technical debt priority; besides, in the current implementation, it is possible to verify the elements involved in the priority classification for each technical debt item.

Loss of information: The information used to classify the TD item priority is available after the prioritization. It is possible to identify the criterion used to define the TD priority.

Rank updates: The current approach implementation enables the rank update whenever any change is performed in the context or the prioritization rule.

Scalability: Currently, the approach implementation already handles hundreds of TD items and dozens of configuration artifacts and IT Assets, with no performance issues (on a small computer instance with limited resources). It is necessary to evaluate if the current implementation can handle thousands of TDs and a higher number of configuration items and value sources.

5.9 Conclusions

We performed an industrial multiple-case study to evaluate how a business prioritization approach for technical debt works in a real scenarios. We observed misalignment regarding the prioritization of technical debt, the value source classification, and the prioritization rule. These conflicts are expected when stakeholders of different domains are involved. Our business-driven approach contributes to the alignment of the business perspectives for technical debt prioritization.

We applied the proposed approach supported by a five-months case study where we associated technical debt items with business-value elements. We also identified and solved conflicting business perspectives among stakeholders. We observed a downward trend in the resolution of technical debt items that are related to high business priority. We also found that much goes on behind the scenes regarding the prioritization decision making. Finally, after a set of interviews and focus groups, we identified eight business factors that affect decision making regarding technical debt.

6 Conclusions

This research has three major results: first, we identified a set of the business causes behind the two most cited causes of technical debt: "tight deadlines" and the "prioritization of features over technical debt payment"; secondly, we developed and evaluated a decision-making approach that prioritizes technical debt considering how IT assets support company's business processes, thus providing a new perspective on technical debt management. Finally, after the experience of applying the approach on a multi-case study on two companies over 11 months, we identified a set of business concerns behind the decision-making on technical debt prioritization. After all, we obtained feedback about the use of the approach.

To the best of our knowledge, our approach to dealing with the technical debt prioritization problem is unique in the way it classifies the business priority of technical debt and how the priority is attributed to them. The business priority is based on the business process management perspective, where we classify and evaluate the impact of the technical debt on a business process (either through key features, user journeys, or the business process itself). A prioritization rule accomplishes the business priority attribution, which may change the prioritization as the business changes.

The results so far contribute to both academia and industry. In academia, the proposed approach creates an opportunity to analyze the conflicts and improve the alignment between technical and business perspectives on technical debt prioritization. The integration between technical debt and other processes and activities can also be explored, like the integration with continuous integration (e.g., identify the business impact of technical debt identified by automatic tools like Sonarqube); requirement engineering (unite the technical debt and requirement prioritization); incident management (improve the technical debt prioritization considering the impact of technical debt on incidents). We also have the opportunity to explore the business and technical management processes to reduce the forces that contribute to technical debt creation.

This research also contributes to industry; The results provide insights about the business perspectives to technical debt prioritization. The proposed approach helps to align the stakeholders involved in technical debt prioritization. The business perspectives from the proposed approach also contribute to the business stakeholders' alignment providing a standard view of the business value, criticity, and feature usage of their products and services. The business causes behind tight deadlines and the prioritization of features over technical debt payment can improve business planning and decision making. The proposed approach and tool can also be extended to integrate with other tools to bring the business perspective to decision-making.

6.1 Contributions and Findings

In this section, we present the current results of our research questions:

RQ: How does the business perspective influence the prioritization of technical debt?

We first asked practitioners how business decisions contribute to technical debt to answer the overall research. We found that business causes are relevant drivers for technical debt (Chapter 2). After, we tested whether a business-driven approach would contribute to technical debt prioritization considering the business process perspective (Chapter 3). Since the result was positive, we worked to build a solution to support the TD prioritization using a business perspective (Chapter 4). Finally, to answer the main research question, we applied a business-driven prioritization approach to two companies and observed how the business perspective influences the TD prioritization (Chapter 5).

We observed that the business-driven technical debt prioritization approach was relevant to:

- Improve the communication about technical debt between the involved stakeholders;
- Align the perception of both business and technical stakeholders about what is core-business or not on their products and services;
- Align the prioritization of the technical debt between business and technical stakeholders;
- Help technical stakeholders convince about technical debt payment using business-related arguments.

We also identified an intricated cause-effect relationship between business, management, and technical causes for technical debt creation. Business aspects play a relevant role in promoting "tight deadlines" and contributing to the prioritization of "features over quality aspects".

In the following sections, we summarize the results of the research questions of each study.

6.1.1 Study 1 - Beyond tight deadlines: what are the business causes of technical debt?

RQ 1.1. How do business decisions contribute to technical debt?

We surveyed 71 respondents and asked them to what extent business decisions lead to the creation of technical debt. 96% of the respondents indicated that business decisions lead to the creation of technical debt (to some extent: 23%; to a great extent: 51%; to a very great extent: 23%) while only 4% indicated no or low influence.

We also asked them to give examples of how business may contribute to technical debt. We analyzed the responses and created a cause-effect model (Figure 2), which relates the various business causes of technical debt to each other and explains their impact on technical debt. Practitioners of different roles can use this model to understand the influences on technical debt creation, anticipate issues, and work across business and IT to better manage technical debt.

6.1.2 Study 2 - Aligning Technical Debt Prioritization with Business Objectives: A Multiple-Case Study

RQ 2.1. How can the business perspective influence the prioritization of technical debt?

We identified a tension between technical and business perspectives regarding the prioritization of technical debt. The lack of alignment between business and technical prioritizations shows how different a purely technical prioritization could be if conducted from a business perspective.

RQ 2.2. Does the business perspective captured through business process management facilitate the prioritization of technical debt?

The IT stakeholders declared that the business process visualization helped prioritize technical debt. They also argued that *"many times a critical technical debt must be prioritized even if it affects a low critical business process, to reduce the problem of accumulating debt"*.

One participant explained that using the business perspective for prioritizing technical debt could also provide an objective way to define policies regarding technical debt.

6.1.3 Study 3 - Tracy: A Business-driven Technical Debt Prioritization Framework

We developed a tool that implements Tracy, the decision-making framework that prioritizes technical debt, considering how IT assets support a company's business processes, thus providing a new perspective on technical prioritization.

The participants evaluated the business approach for technical debt prioritization. They agreed on the approach's usefulness and endorsed the new perspective on making prioritization decisions. They agreed that it could be easier to argue with managers and customers about the prioritization of critical technical debt with the business information related to technical debt. They also liked the idea of having a standard set of business metrics to define "what is important" in terms of which technical debt should be prioritized.

Our proposed approach to technical debt prioritization considers a balance between simplicity and effectiveness for decision making. Different from the other existing research on technical debt prioritization, our method to measure the business impact of technical debt is based on three entities:

- the technical artifacts (e.g., system, service, module) being affected by the technical debt;
- the products and services supported by the technical artifacts;
- the business processes and critical features that create business value.

6.1.4 Study 4 - Business-driven Technical Debt Prioritization: a Multiple Case Study

RQ 4.1: How does the proposed business-driven approach impact technical debt prioritization?

We confirmed the first study's finding where: the contrast between the technical and business-driven prioritization shows that a business approach contributes to the alignment between technical and business decision making. We also observed that technical debt with high business priority received more attention to be paid during both case studies. In the first case, after the business-driven technical debt prioritization approach, the trend to pay high-business impact technical debt increased.

RQ 4.2: What are the business stakeholder's perceptions regarding factors that should influence technical debt prioritization?

After analyzing the participants' arguments during the meetings to discuss the value source classification and the prioritization rule, we found eight factors that should influence the technical prioritization, considering the business perspective:

- core-business value sources;
- usage frequency;
- risk perception;
- time negotiation;
- delivery frequency;
- embedded systems targeted by regulation and certification;
- incidents;
- the perception of marketing versus functional perception.

RQ 4.3: What are the benefits and potential improvements of running the approach in a real scenario?

We observed that the business-driven technical debt approach regularly supported the prioritization decision-making (during sprint plannings) on both case studies. As reported by participants during the final interview, the prioritization approach also contributed to the communication and alignment between technical and business stakeholders. The concepts and data used in the approach have been shown to be adequate and feasible to collect and maintain in both cases (in both cases, the data collection was straightforward to collect and maintain).

6.2 Opportunities for Future Research

The current results open other research related to business-driven technical debt management.

6.2.1 Replication of the case studies

Since our results are supported by case studies, the application of our approach to new environments contributes to its evolution and generalization. We had the opportunity to work with a startup that did not yet release its product. We will continue monitoring our approach to observe the changes in the technical debt prioritization once the system's states change from to-be operational to operational.

6.2.2 Investigate the business causes behind technical debt

We did a first step toward analyzing business causes behind tight deadlines and the prioritization of features over quality aspects. This study opens the opportunity to study how the decision processes from the business stakeholders can be enhanced to improve the alignment between business and IT areas and reduce the unnecessary forces that contribute to the creation of technical debt.

6.2.3 “Back to the future” business prioritization

The current approach can be evolved to support the prioritization of technical debt with low priority in the present but high priority in the future. Technical debt that needs high effort and time to be paid, like architectural debt, must have a way to be prioritized so that the team can anticipate the effort to pay the debt.

6.2.4 Smart technical debt payment prioritization

In this research, we observed that deciding what technical debt must be selected to be paid involves a more extensive set of variables than the business priority. Future work can explore machine learning techniques that connect requirements, incidents, the business perspective, and other variables to suggest the technical debt that must be paid to reduce business risk.

6.2.5 Risk analysis

During the last case study, we implemented the support to the user register the chance of the registered technical debt to affect different aspects of the products and services. E.g., it is possible to classify if a technical debt has a (high, medium, or low) chance to provoke an incident, affect the system’s performance, etc. The current approach must be extended to support this and other types of risk analysis.

6.2.6 Measure the technical debt cost on the team’s routine

During some observations on planning poker meetings, we verified that technical debt creates a hidden cost on the team’s routine. Like when someone delays the implementation of a feature due to technical debt. The friction created by technical debt can be studied. Technical debt that creates relevant friction could be prioritized, even if it does not affect systems with high business value.

6.2.7 Integrate TDM with incident management

The TD prioritization and impact measurement can be enhanced by integrating the technical debt management process with IT infrastructure processes, e.g., incident management, change management, service support.

6.2.8 Add business value to coding environment

We also expect to create a solution that offers a business perspective to the developer's coding environment. We expect to create an environment for the developer to be aware of the business impact of the code that she is working on. We expect that the awareness of business aspects can prevent the creation of technical debt on the code that has a relevant business impact.

Bibliography

[Abad, Conference e 2015]ABAD, Z. S. H.; CONFERENCE, G. R. R. E.; 2015. Using real options to manage technical debt in requirements engineering. *ieeexplore.ieee.org*, IEEE, p. 230–235. ISSN 978-1-4673-6905-3.

[Akbarinasaji 2015]AKBARINASAJI, S. Toward measuring defect debt and developing a recommender system for their prioritization. *CEUR Workshop Proceedings*, v. 1469, p. 15–20, 2015. ISSN 16130073.

[Al-Barak e Bahsoon 2018]AL-BARAK, M.; BAHSOON, R. Prioritizing technical debt in database normalization using portfolio theory and data quality metrics. In: *2018 International Conference on Technical Debt*. [S.l.]: ACM Press, 2018. p. 31–40. ISBN 9781450357135.

[Aldaeej e Seaman 2018]ALDAEEJ, A.; SEAMAN, C. B. From lasagna to spaghetti, a decision model to manage defect debt. In: *2018 International Conference on Technical Debt*. [S.l.: s.n.], 2018.

[Alfayez et al. 2020]ALFAYEZ, R. et al. A Systematic Literature Review of Technical Debt Prioritization. In: *International Conference on Technical Debt TechDebt*. [S.l.: s.n.], 2020. p. 1–10.

[Allman 2012]ALLMAN, E. Managing technical debt. *Commun. ACM*, v. 55, n. 5, p. 50–55, 2012.

[Almeida, Treude e Kulesza]ALMEIDA, R. R. de; TREUDE, C.; KULESZA, U. *Companion Data*.

[Alves et al. 2016]ALVES, N. S. et al. Identification and management of technical debt. *Inf. Softw. Technol.*, Butterworth-Heinemann, v. 70, n. C, p. 100–121, fev. 2016. ISSN 0950-5849.

[Ampatzoglou et al. 2015]AMPATZOGLOU, A. et al. The financial aspect of managing technical debt: A systematic literature review. *Information and Software Technology*, v. 64, p. 52–73, 2015. ISSN 0950-5849.

[Avgeriou et al. 2016]AVGERIOU, P. et al. Managing Technical Debt in Software Engineering (Dagstuhl Seminar 16162). *Dagstuhl Reports*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, v. 6, n. 4, p. 110–138, 2016. ISSN 2192-5283.

[Bourque, Fairley e Society 2014]BOURQUE, P.; FAIRLEY, R. E.; SOCIETY, I. C. *Guide to the Software Engineering Body of Knowledge (SWEBOK(R)): Version 3.0*. 3rd. ed. [S.l.]: IEEE Computer Society Press, 2014. ISBN 0769551661.

[Braun e Clarke 2006]BRAUN, V.; CLARKE, V. Using thematic analysis in psychology. *Qualitative Research in Psychology*, Routledge, v. 3, n. 2, p. 77–101, 2006.

[Brennan e Prediger 1981]BRENNAN, R. L.; PREDIGER, D. J. Coefficient kappa: Some uses, misuses, and alternatives. *Educational and Psychological Measurement*, v. 41, n. 3, p. 687–699, 1981.

[Buschmann 2011]BUSCHMANN, F. To pay or not to pay technical debt. *IEEE Software*, v. 28, n. 6, p. 29–31, Nov 2011. ISSN 0740-7459.

[Caddick e Cable 2011]CADDICK, R.; CABLE, S. *Communicating the User Experience: A Practical Guide for Creating Useful UX Documentation*. [S.l.]: Wiley, 2011. ISBN 9781119971108.

[Choudhary e Singh 2016]CHOUDHARY, A.; SINGH, P. Minimizing refactoring effort through prioritization of classes based on historical, architectural and code smell information. *CEUR Workshop Proceedings*, v. 1771, n. Tda, p. 76–79, 2016. ISSN 16130073.

[Codabux e Williams 2016]CODABUX, Z.; WILLIAMS, B. J. Technical debt prioritization using predictive analytics. In: *the 38th International Conference*. [S.l.]: IEEE-CS DATC, IEEE Computer Society, 2016. (ICSE '16), p. 704–706. ISBN 978-1-4503-4205-6.

[Cunningham 1992]CUNNINGHAM, W. The wycash portfolio management system. *SIGPLAN OOPS Mess.*, ACM, v. 4, n. 2, p. 29–30, dez. 1992. ISSN 1055-6400.

[Dumas Marcello La Rosa e Reijers 2018]DUMAS MARCELLO LA ROSA, J. M. M.; REIJERS, H. *Fundamentals of Business Process Management*. 2. ed. [S.l.]: Springer-Verlag Berlin Heidelberg, 2018.

[Fernández-Sánchez et al. 2017]FERNÁNDEZ-SÁNCHEZ, C. et al. Identification and analysis of the elements required to manage technical debt by means of a systematic mapping study. *Journal of Systems and Software*, v. 124, p. 22–38, 2017. ISSN 0164-1212.

[Fontana et al. 2015]FONTANA, F. A. et al. Towards a prioritization of code debt: A code smell intensity index. In: *2015 IEEE 7th International Workshop on Managing Technical Debt (MTD)*. [S.l.: s.n.], 2015. p. 16–24.

[Freire et al. 2021]FREIRE, S. et al. How Experience Impacts Practitioners' Perception of Causes and Effects of Technical Debt. In: *14th International Conference on Cooperative and Human Aspects of Software Engineering (CHASE 2021)*. [S.l.: s.n.], 2021.

[Given 2008]GIVEN, L. M. (Ed.). *The SAGE Encyclopedia of Qualitative Research Methods*. [S.l.]: SAGE, 2008.

[Guo e Seaman 2011]GUO, Y.; SEAMAN, C. A portfolio approach to technical debt management. In: *Proceedings of the 2nd Workshop on Managing Technical Debt*. [S.l.]: ACM, 2011. (MTD '11), p. 31–34. ISBN 978-1-4503-0586-0.

[Guo et al. 2011]GUO, Y. et al. Tracking technical debt; an exploratory case study. In: *2011 27th IEEE International Conference on Software Maintenance (ICSM)*. [S.l.: s.n.], 2011. p. 528–531. ISSN 1063-6773.

[Guo, Spínola e Seaman 2016]GUO, Y.; SPÍNOLA, R. O.; SEAMAN, C. Exploring the costs of technical debt management – a case study. *Empirical Software Engineering*, v. 21, n. 1, p. 159–182, 2016. ISSN 1573-7616.

[Gupta et al. 2016]GUPTA, R. K. et al. Pragmatic approach for managing technical debt in legacy software project. In: *Proceedings of the 9th India Software Engineering Conference*. [S.l.]: Association for Computing Machinery, 2016. (ISEC '16), p. 170–176. ISBN 9781450340182.

[Harun e Lichter 2015]HARUN, M. F.; LICHTER, H. Towards a Technical Debt Management Framework based on Cost-Benefit Analysis. In: *10th International Conference on Software Engineering Advances*. [S.l.: s.n.], 2015.

[Kniberg e Ivarsson 2012]KNIBERG, H.; IVARSSON, A. *Scaling agile at Spotify with Tribes, Squads, Chapters & Guilds*. 10 2012.

[Kruchten, Nord e Ozkaya 2019]KRUCHTEN, P.; NORD, R.; OZKAYA, I. *Managing Technical Debt: Reducing Friction in Software Development*. [S.l.]: Software Engineering Institute, Carnegie Mellon University, 2019.

[Kruchten et al. 2012]KRUCHTEN, P. et al. Technical debt in software development: From metaphor to theory report on the third international workshop on managing technical debt. *SIGSOFT Softw. Eng. Notes*, ACM, v. 37, n. 5, p. 36–38, set. 2012. ISSN 0163-5948.

[Lemon e Verhoef 2016]LEMON, K. N.; VERHOEF, P. C. Understanding customer experience throughout the customer journey. *Journal of Marketing*, v. 80, n. 6, p. 69–96, 2016.

[Lenarduzzi et al. 2019]LENARDUZZI, V. et al. Technical debt prioritization: State of the art. A systematic literature review. *CoRR*, abs/1904.12538, 2019.

[Letouzey e Ilkiewicz 2012]LETOUZEY, J.-L.; ILKIEWICZ, M. Managing technical debt with the sqale method. *IEEE Software*, v. 29, n. 6, p. 44–51, 2012.

[Lim, Taksande e Seaman 2012]LIM, E.; TAKSANDE, N.; SEAMAN, C. A balancing act: What software practitioners have to say about technical debt. *IEEE Software*, v. 29, n. 6, p. 22–27, Nov 2012. ISSN 0740-7459.

[Marciuska, Gencel e Abrahamsson 2013]MARCIUSKA, S.; GENCEL, C.; ABRAHAMSSON, P. Exploring how feature usage relates to customer perceived value: A case study in a startup company. In: HERZWURM, G.; MARGARIA, T. (Ed.). *Software Business. From Physical Products to Software Services and Solutions*. [S.l.]: Springer Berlin Heidelberg, 2013. p. 166–177. ISBN 978-3-642-39336-5.

[Martini, Besker e Bosch 2018]MARTINI, A.; BESKER, T.; BOSCH, J. Technical debt tracking: Current state of practice: A survey and multiple case study in 15 large organizations. *Science of Computer Programming*, v. 163, p. 42–61, 2018. ISSN 0167-6423.

[Martini e Bosch 2015]Martini, A.; Bosch, J. Towards prioritizing architecture technical debt: Information needs of architects and product owners. In: *2015 41st Euromicro Conference on Software Engineering and Advanced Applications*. [S.l.: s.n.], 2015. p. 422–429.

[Mensah et al. 2018]MENSAH, S. et al. On the value of a prioritization scheme for resolving Self-admitted technical debt. *Elsevier*, v. 135, p. 37–54, jan 2018.

[OMG 2011]OMG. *Business Process Model and Notation (BPMN), Version 2.0*. Jan 2011.

[Plösch et al. 2018]PLÖSCH, R. et al. Design debt prioritization - a design best practice-based approach. In: *2018 International Conference on Technical Debt*. [S.l.: s.n.], 2018.

[Ramasubbu e Kemerer 2019]Ramasubbu, N.; Kemerer, C. F. Integrating technical debt management and software quality management processes: A normative framework and field tests. *IEEE Transactions on Software Engineering*, v. 45, n. 3, p. 285–300, 2019. ISSN 0098-5589.

[Rebouças de Almeida et al. 2018]Rebouças de Almeida, R. et al. Aligning technical debt prioritization with business objectives: A multiple-case study. In: *Proc. of the Int'l. Conf. on Software Maintenance and Evolution - ICSME*. [S.l.: s.n.], 2018. p. 655–664. ISSN 2576-3148.

[Rebouças de Almeida et al. 2021]Rebouças de Almeida, R. et al. Business-driven technical debt prioritization: an industrial case study. In: *Proc. of the Int'l. Conf. on Technical Debt - TechDebt2021*. [S.l.: s.n.], 2021.

[Rebouças de Almeida, Treude e Kulesza 2019]Rebouças de Almeida, R.; Treude, C.; Kulesza, U. Tracy: A business-driven technical debt prioritization framework. In: *Proc. of the Int'l. Conf. on Software Maintenance and Evolution - ICSME*. [S.l.: s.n.], 2019.

[Ribeiro et al. 2016]RIBEIRO, L. F. et al. Decision criteria for the payment of technical debt in software projects: A systematic mapping study. In: *Proc. of the Int'l. Conf. on Enterprise Information Systems*. [S.l.: s.n.], 2016. p. 572–579. ISBN 978-989-758-187-8.

[Ribeiro et al. 2017]Ribeiro, L. F. et al. A strategy based on multiple decision criteria to support technical debt management. In: *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. [S.l.: s.n.], 2017. p. 334–341.

[Rios, Neto e Spínola 2018]RIOS, N.; NETO, M. G. de M.; SPÍNOLA, R. O. A tertiary study on technical debt: Types, management strategies, research trends, and base information for practitioners. *Information and Software Technology*, v. 102, p. 117–145, 2018. ISSN 0950-5849.

[Rios et al. 2018]Rios, N. et al. A study of factors that lead development teams to incur technical debt in software projects. In: *Proc. of the Euromicro Conf. on Software Engineering and Advanced Applications*. [S.l.: s.n.], 2018. p. 429–436.

[Rios et al. 2020]RIOS, N. et al. 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*, v. 25, 09 2020.

[Rosing August-Wilhelm Scheer 2015]ROSING AUGUST-WILHELM SCHEER, H. v. S. Mark von. *The Complete Business Process Handbook*. [S.l.]: Elsevier, 2015.

[Runeson e Höst 2008]RUNESON, P.; HÖST, M. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, v. 14, n. 2, p. 131, Dec 2008. ISSN 1573-7616.

[Runeson et al. 2012]RUNESON, P. et al. *Case Study Research in Software Engineering: Guidelines and Examples*. 1st. ed. [S.l.]: Wiley Publishing, 2012. ISBN 1118104358.

[Sae-Lim, Hayashi e Saeki 2016]SAE-LIM, N.; HAYASHI, S.; SAEKI, M. Context-based code smells prioritization for prefactoring. In: *2016 IEEE 24th International Conference on Program Comprehension (ICPC)*. [S.l.: s.n.], 2016. p. 1–10.

[Schmid 2013]SCHMID, K. A Formal Approach to Technical Debt Decision Making. In: *Proceedings of the 9th International ACM Sigsoft Conference on Quality of Software Architectures*. [S.l.]: Association for Computing Machinery, 2013. (QoSA '13), p. 153–162. ISBN 9781450321266.

[Seaman e Guo 2011]SEAMAN, C.; GUO, Y. Measuring and monitoring technical debt. In: ZELKOWITZ, M. V. (Ed.). [S.l.]: Elsevier, 2011. (Advances in Computers, v. 82), p. 25–46. ISSN 0065-2458.

[Seaman et al. 2012]SEAMAN, C. et al. Using technical debt data in decision making: Potential decision approaches. In: *2012 Third International Workshop on Managing Technical Debt (MTD)*. [S.l.: s.n.], 2012. p. 45–48.

[Silver 2011]SILVER, B. *BPMN Method and Style*. 2. ed. [S.l.]: Cody-Cassidy Press, 2011.

[Snipes et al. 2012]SNIPES, W. et al. Defining the decision factors for managing defects: A technical debt perspective. In: *Proceedings of the Third International Workshop on Managing Technical Debt*. [S.l.]: IEEE Press, 2012. (MTD '12), p. 54–60. ISBN 978-1-4673-1749-8.

[Storey et al. 2017]Storey, M. et al. Using a visual abstract as a lens for communicating and promoting design science research in software engineering. In: *Proc. of the Int'l. Symp. on Empirical Software Engineering and Measurement*. [S.l.: s.n.], 2017. p. 181–186.

[Tornhill 2018]TORNHILL, A. Prioritize technical debt in large-scale systems using codescene. In: *the 2018 International Conference*. [S.l.]: SIGSOFT, ACM Special Interest Group on Software Engineering, 2018. p. 59–60. ISBN 9781450357135.

[UK 2012]UK itSMF. *ITIL Foundation Handbook*. 3rd. ed. [S.l.]: The Stationery Office, 2012. ISBN 0113313497.

[Verdecchia, Kruchten e Lago 2020]VERDECCHIA, R.; KRUCHTEN, P.; LAGO, P. Architectural technical debt: A grounded theory. In: JANSEN, A. et al. (Ed.). *Software Architecture*. Cham: Springer International Publishing, 2020. p. 202–219. ISBN 978-3-030-58923-3.

[Vidal et al. 2015]VIDAL, S. et al. Jspirit: a flexible tool for the analysis of code smells. In: *2015 34th International Conference of the Chilean Computer Science Society (SCCC)*. [S.l.: s.n.], 2015. p. 1–6.

[Wieringa 2014]WIERINGA, R. J. *Design Science Methodology for Information Systems and Software Engineering*. 1. ed. [S.l.]: Springer-Verlag Berlin Heidelberg, 2014.

[Wolff e Johann 2015]WOLFF, E.; JOHANN, S. Technical debt. *IEEE Software*, v. 32, n. 4, p. 94–c3, July 2015. ISSN 0740-7459.

[Yin 2018]YIN, R. K. *Case Study Research and Applications: Design and Methods*. 6. ed. [S.l.]: SAGE Publishing, 2018.

[Yli-Huumo, Maglyas e Smolander 2016]YLI-HUUMO, J.; MAGLYAS, A.; SMOLANDER, K. How do software development teams manage technical debt? – an empirical study. *Journal of Systems and Software*, v. 120, p. 195–218, 2016. ISSN 0164-1212.

[Yli-Huumo et al. 2015]YLI-HUUMO, J. et al. The relationship between business model experimentation and technical debt. In: ______. *Software Business: 6th International Conference, ICSOB 2015, Braga, Portugal, June 10-12, 2015, Proceedings*. [S.l.]: Springer International Publishing, 2015. p. 17–29. ISBN 978-3-319-19593-3.

[Zazworka, Seaman e Shull 2011]ZAZWORKA, N.; SEAMAN, C.; SHULL, F. Prioritizing design debt investment opportunities. In: *Proceedings of the 2nd Workshop on Managing Technical Debt*. [S.l.]: Association for Computing Machinery, 2011. (MTD '11), p. 39–42. ISBN 9781450305860.