University of Brasília

Institute of Exact Sciences
Department of Computer Science

A Socio-Technical Case Study on Integrating Product Management and DevOps Practices in Governmental Software Delivery

Bruna Pinos de Oliveira

Thesis presented in partial fulfillment of the requirements for the degree of Master of Science in Informatics

Advisor
Prof.a Dr.a Carla Silva Rocha Aguiar

Brazil
2026

University of Brasília

Institute of Exact Sciences
Department of Computer Science

A Socio-Technical Case Study on Integrating Product Management and DevOps Practices in Governmental Software Delivery

Bruna Pinos de Oliveira

Thesis presented in partial fulfillment of the requirements for the degree of Master of Science in Informatics

Prof.a Dr.a Carla Silva Rocha Aguiar (Advisor)
UnB

Genaina Nunes Rodrigues	Paulo Roberto Miranda Meirelles
UnB	USP

Rodrigo Bonifácio de Almeida
UnB

Prof.a Dr.a Cláudia Nalon
Coordinator of the Graduate Program in Informatics

Brazil, Brasília, March 31 2026

Acknowledgements

I am deeply thankful to my family for their support throughout this journey. I also thank my advisor, Professor Carla Silva Rocha Aguiar, for her guidance, encouragement, and valuable contributions during my master's research.

I am grateful to everyone at the UnB LabLivre laboratory, especially those who supported me and took part in the Brasil Participativo project. Likewise, I appreciate the collaboration and contributions from the Directorate of Social Participation of the General Secretariat of the Presidency of the Republic.

This study was financed in part by the General Secretariat of the Presidency of the Republic within the scope of the Brasil Participativo Project, and also in part by the Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil (CNPq) - Process number 131789/2024-4.

Abstract

Context: Continuous Delivery has become a central capability for software-producing organizations, enabling frequent and reliable delivery of value. While Agile, DevOps, and Software Product Management each contribute practices and perspectives to support Continuous Delivery, research often addresses these areas in silos.

Objectives: This study investigates DevOps practices and Software Product Management challenges in a real-world co-development project between academia and government, the Brasil Participativo Digital Participation Platform. Specifically, it seeks to identify team structures and practices that support effective collaboration and continuous delivery in such complex, multi-stakeholder settings.

Methods: The study adopts an exploratory case study approach, using the Brasil Participativo project as the empirical setting. We applied directed content analysis, using pre-existing taxonomies of DevOps practices and Software Product Management problems. We collected empirical evidence from project artifacts, documentation, and collaboration tools, and analyzed them to assess the presence of the practices or problems.

Results: The results show that team structures and practices directly impact product management effectiveness. Phases that included practices fostering integrated teams, defined workflows, and continuous alignment sessions demonstrated greater adoption of DevOps practices and fewer Product Management problems. The presence of a Product Manager/Owner helps teams adhere to the product view, collaboratively create roadmaps, prioritize requirements, and maintain communication between government and academic teams.

Keywords: Continuous Delivery, DevOps, Product Management, Team Structure, E-government, Co-development

Resumo

Título: Integrando Gerenciamento de Produto e Práticas DevOps em Entregas de Software Governamentais: Um Estudo de Caso Socio-Técnico do Brasil Participativo.

Contexto: A Entrega Contínua (Continuous Delivery) é uma prática importante para organizações que produzem software, permitindo a entrega frequente e confiável de valor. Embora o Ágil, o DevOps e a Gestão de Produtos de Software (Software Product Management) contribuam, cada um, com práticas e perspectivas para apoiar a Entrega Contínua, essas áreas são frequentemente abordadas de forma isolada em estudos.

Objetivos: Esta pesquisa visa investigar as práticas de DevOps e os problemas de Gestão de Produtos de Software, e compreender como estes impactam o desenvolvimento da plataforma de participação social Brasil Participativo.Esta plataforma foi concebida em um ambiente de desenvolvimento colaborativo entre o governo e a academia. O objetivo é identificar estruturas e práticas de equipe que apoiem a colaboração efetiva entre equipes em projetos de colaboração multissetorial.

Métodos: O estudo adota uma abordagem de estudo de caso exploratório, com o projeto Brasil Participativo como objeto de estudo. Aplica-se uma análise de conteúdo direcionada, valendo-se de taxonomias pré-existentes de práticas de DevOps e de problemas de gestão de produtos de software. Evidências empíricas são coletadas a partir de artefatos do projeto, documentação e ferramentas de colaboração, e analisadas sistematicamente para avaliar a presença e a interação desses elementos.

Resultados: Os resultados mostram que as estruturas de equipe e as práticas impactam diretamente a eficácia da gestão de produtos. As fases que incluíram práticas que promoviam reuniões de equipes, fluxos de trabalho definidos e sessões de alinhamento contínuo demonstraram maior adoção de práticas de DevOps e menos problemas de Gestão de Produto. A presença de um Gerente/Dono de Produto (Product Manager/Owner) ajuda as equipes a aderirem à visão do produto, a criarem roadmaps de forma colaborativa, a priorizarem requisitos e a manterem a comunicação entre as equipes do governo e da academia.

Palavras-chave: Entrega Contínua, DevOps, Gerenciamento de Produto, Estrutura de times, E-government, Colaboração multissetorial

Contents

1 Introduction **1**
1.1 Problem outline 1
1.2 Research Questions 3
1.3 Research Design 4
1.4 Claimed Contributions 6
1.5 Thesis Structure 6

2 Background **8**
2.1 Continuous Delivery 8
2.2 Product Management 9
2.2.1 Agile Practices 12
2.2.2 DevOps Practices 13
2.3 Co-development and Government software 16
2.4 Technopolitics and Technopolitical Softwares 18
2.4.1 Brasil Participativo 19
2.4.2 Decidim 21

3 Research Design **23**
3.1 Case Study 24
3.1.1 Data Collection 26
3.1.2 Data Analysis 27
3.1.3 Plan Validity 29

4 Brasil Participativo **32**

5 Results **38**
5.1 Data Analysis 41
5.2 Focus-Group Validation 42
5.3 Phase 1: PPA and Early Experiences (March to December 2023) 44
5.4 Phase 2 - January 2024 to February 2025 48

5.5 Phase 3 (February 2025 – March 2026) 52
5.6 Discussion 55

6 Conclusion **66**
6.1 Future work 67
6.2 Thread to Validity 67

7 Appendices **69**
7.1 Appendix - A. DevOps practices Codebook 70
7.2 Appendix - B. Product Management Problems - Final Evaluation 71
7.3 Appendix - C. DevOps Practices - Final Evaluation 72
7.4 Appendix-D. Case Study Evidences 74
7.5 Appendix - E. Brasil Participativo Team Characterization according to the Focus Group Perception 75

References **76**

List of Figures

1.1 The Case study steps followed for this study. 5

2.1 Product management in the entire product-life cycle (Ebert and Brinkkemper, 2014). 10
2.2 Full representation of the DevOps Constructs and Proposition from Díaz et al. (2024) theory 16
2.3 5 types of participation processes in Brazil Participation and Decidim components for building the processes 20
2.4 Decidim's functional architecture, showing a combination of components in participatory spaces (Barandiaran et al., 2024). 22

3.1 The Case study steps followed for this study. (show also in section 1.3) . . 24
3.2 Data Analysis Stage 1 - Method to determine occurrence of the DevOps Codes and Product Management Problem in each Phase of the project (see Figure 3.1) 29
3.3 Data Analysis Stage 2 - Inputs of Focus Group sessions and Results (see Figure 3.1) 30

4.1 Composition of the multissectorial arrangement of the Brasil Participativo project (Silva Rocha Aguiar et al., 2024). 33
4.2 Opinion Survey applied in G20 Participativo Process to fast participation engagement 35
4.3 Directory and LabLivre members present in the 4th National Youth Conference 36
4.4 Actual Home Page of Brasil Participativo platform (as of March 2026) . . . 37

5.1 Roles and Communication across Stakeholders 40
5.2 Example of DevOps practices content analysis for Brasil Participativo in Phase 1 41

5.3 Comparison of the quantity of Product Management occurrences. Especially in phase 3, the focus group strongly identified the presence of more problems. 43
5.4 Comparison of the quantity of DevOps practices occurrences. In phase 2, the researcher obtained an optimistic perception, identifying more practices when the focus group classified them as not present. 44
5.5 Excerpt from the Integrated Activities Method diagram - Phase 1 47
5.6 The quantity of DevOps practices and SPM problems identified throughout the project, divided into phases. Phase 2 shows the most identified problems and the lack of DevOps practices. 60
5.7 Relationship between DevOps Practices and Team Structures of Phase 1 - adapted from Díaz et al. (2024) and with indicatives of presence based on the study analysis 61
5.8 Relationship between DevOps Practices and Team Structures of Phase 2 - adapted from Díaz et al. (2024) and with indicatives of presence based on the study analysis 62
5.9 Relationship between DevOps Practices and Team Structures of Phase 3 - adapted from Díaz et al. (2024) and with indicatives of presence based on the study analysis 62

List of Tables

2.1 Software Product Management Problems (Adapted from Springer and Miler (2022)). 12

3.1 Present Case Study Protocol 25

5.1 Example of the Evidence Categorization - Full table can be found in Appendix-D (section 7.4) 41
5.4 Phase 1 - Summary of content analysis results. 45
5.2 Government team composition throughout Phases - It did not change in the duration of the project 46
5.3 Academia team composition in Phase 1 46
5.5 Phase 1 - Teams and Artifacts 48
5.6 Academia team composition in Phase 2 49
5.7 Phase 2 - Summary of content analysis results. 49
5.8 Comparison of Roles across Phases 1 and 2 52
5.9 Comparison of Artifacts across Phases 1 and 2 52
5.10 Academia team composition in Phase 3 53
5.11 Phase 3 - Summary of content analysis results. 54
5.12 Comparison of Roles across Phase 1, 2, and 3 55
5.13 Comparison of Artifacts across Phases 1, 2, and 3 56
5.14 Improved Product Management Problems at the end of the Project 56
5.15 DevOps Practices present in all Phases 57
5.16 DevOps Practices that have worsened over time 57
5.17 Product Management Problems that persisted throughout the Project . . . 61
5.18 Recommended Team Structure to improve Product Management in Government-Academia Software Co-development 64
5.19 Recommended Artifacts and Practices to improve Product Management in Government-Academia Software Co-development 65

7.1 The DevOps Practice codebook used in this research was adapted from Díaz et al. (2024). Original codebook available in the supplementary material. . 70
7.2 Final Evaluation Product Management Problems - Full table with Researcher's note, and Focus group quotations are available in the supplementary material. 71
7.3 Final Evaluation of DevOps Practices - Full table with Researcher's note, and Focus group quotations are available in the supplementary material. . 72
7.4 All Evidence Collected in this research - Full table with description and links are available in the supplementary material 74

Chapter 1

Introduction

Developing e-government software from a project management perspective differs from other ICT projects due to its complexity and large scale, as it combines innovation, politics, and social impacts and spans a wide scope, target audience, and organizational size (Anthopoulos et al., 2016; Wen et al., 2020).

One promising approach to fostering innovation in public sector software is co-development between government and academia. In this model, both parties collaborate across the full lifecycle of a technology, from design and creation through to continuous improvement. Academia's contribution is particularly valuable in this arrangement, as it brings cutting-edge development practices and advanced project management methodologies to the table. This matters because poor project management is consistently cited as one of the leading causes of e-governance project failure (Anthopoulos et al., 2016), and academic involvement has been shown to meaningfully increase the likelihood of project success (Wen et al., 2020).

This research presents a case study examining academia's contribution to the development of an e-governance software project, Brasil Participativo, through the lens of product management. By observing a complete co-development project from end to end, we identify the product management challenges that arose and the practices that sustained continuous value delivery. From this analysis, we derive a set of best practices, team structures, and artifacts to strengthen government-academia collaboration and build digital products that endure beyond the project's timeline.

1.1 Problem outline

E-government, by definition, "is the use of information technology to support government operations, engage citizens, and provide government services" (Scholl, 2003). Anthopoulos et al. (2016) notes that the implementation of an e-government project has particularities arising from its complexity, including organizational size, resistance to change, novelty, impact on end users, and politics. One of the problems with the failure to implement these projects is poor project management, including weak requirements definition, lack of focus, ambiguous business needs, and unclear vision.

Project management and product management are related but distinct disciplines. While a project is a temporary endeavor undertaken to create a specific output (Ebert and Brinkkemper, 2014), a product is typically developed continuously over a longer timescale, evolving through multiple projects (Springer and Miler, 2022). Product management, accordingly, places greater emphasis on customers, user feedback, and sustained value growth. Given that government-academia collaborations operate within fixed timeframes and are vulnerable to external pressures, including political changes, framing e-government initiatives as products rather than projects can improve their prospects for long-term impact.

Tensions between product management and development teams are well-documented, with common friction points including requirements prioritization, backlog deviations, and implementation disputes (Kittlaus and Fricker, 2017; Springer and Miler, 2022). Through a systematic literature review and survey-based research, Springer and Miler (2022) identifies 27 recurring product management problems and their perceived severity, among them insufficient continuous integration and delivery, frequent priority shifts, and siloed ways of working.

One organizational response to these challenges is the adoption of Continuous Delivery, whereby teams release high-quality software in short cycles and maintain the capacity for reliable deployment at any time (Chen, 2015). This approach yields faster user feedback, higher product quality, and greater innovation capacity (Chen, 2015; Díaz et al., 2021). Since it is common for the government staff to expect and be prepared to validate and deploy a single deliverable, but due to the large characteristics of an e-goverment, with various features development, a continuous delivery method with automation and monitoring tools is a suitable approach and can also help overcome mistrust in a co-development project (Siqueira et al., 2018).

A key enabler of Continuous Delivery in practice is DevOps, a collaborative, multi-disciplinary effort within an organization to automate and sustain continuous delivery pipelines (Leite et al., 2019). Rooted in agile principles, DevOps extends agile methodology by proposing practices specifically designed to enable effective short delivery cycles (Leite et al., 2019). Adopting DevOps, however, requires more than upgrading software tooling: it demands new ways of integrating teams across an organization, since releasing features at pace inherently involves coordination across boundaries. Leite et al. (2020); López-Fernández et al. (2022) examine the organizational structures that emerge in teams practicing continuous delivery and the implications these structures have on delivery performance. Complementing this, Luz et al. (2018, 2019) investigates DevOps adoption in practice and recommends concrete paths forward, identifying collaboration as the central concern underlying successful adoption. Building on this body of work, Díaz et al. (2024) consolidates existing knowledge into coherent taxonomies encompassing DevOps practices, organizational structures, and team characteristics.

In Meirelles et al. (2017); Silva Rocha Aguiar et al. (2024); Siqueira et al. (2018); Wen et al. (2018, 2020), they highlight the strengths of developing e-government through software collaborative development (**co-development**) in partnership between government and academia. Academia offers greater flexibility, working on cutting-edge technology and methods, supported by the government, which uses more traditional techniques but can provide infrastructure, convene stakeholders, and distribute funds, supporting high-risk foundations for innovation (Tanaka and Lopez, 2023; Wen et al., 2020).

Although collaborative development can minimize risk and reduce costs and time-to-market (Büyüközkan and Arsenyan, 2012), the cultures, development methods, and workflows of the two complex organizations can conflict, leading to problems in project management (Wen et al., 2020). The use of agile and FLOSS (Free-Libre-Open Source Software) practices to harmonize the project management aspects and mitigate the cultural gap between organizations in a government-academia project was extensively described by Wen et al. (2018, 2020), by observing the application of this practice in the development of the SPB (Brazilian Public Software) platform.

This thesis extends the work of Wen et al. (2018, 2020) by examining the product management aspect of the government-academia collaboration project. Instead of harmonizing the institution's different project management cultures through FLOSS and agile practices, this study examines the product management problems that arise during the co-development of an e-government product, drawing on the product management common problems in industry described by Springer and Miler (2022).

1.2 Research Questions

This research is guided by the following research questions:

RQ1. How do different continuous delivery practices and team configurations affect the product management of a software in a government-academia collaboration context?

This question examines, across the 3 phases of Brasil Participativo projects, the intersection between product management challenges and continuous delivery practices, investigating how different team configurations and practices address or create specific product management problems in the context of collaborative e-government software development.

The hypothesis regarding this question is:

Hypothesis: Low adoption of continuous delivery best practices can lead to more product management problems, and specific artifacts and team structures can help ensure the applicability of these practices.

RQ2. What practices and team structures favor the development of a government-academia collaborative product?

Based on the practices and product management problems identified, the thesis discusses and proposes alternatives for managing a product in this context to ensure continuous value delivery. Following (Wen et al., 2018), we focus on best practices that helped harmonize interactions between two organizations and their development processes to support successful collaboration.

1.3 Research Design

This research employs a qualitative case study approach, following the methodological framework of the 4 steps: 1. Case study definition and plan, 2. Data collection, 3. Data Analysis and 4. Report (Wohlin et al., 2012; Yin, 2018). As shown in Figure 1.1, the first step is to define the case study goal and the case object. We investigate the Brasil Participativo project (case object) to examine how team structures and product management practices intersect in government-academia collaborative software development and to answer the 2 Research questions defined in Section 1.2.

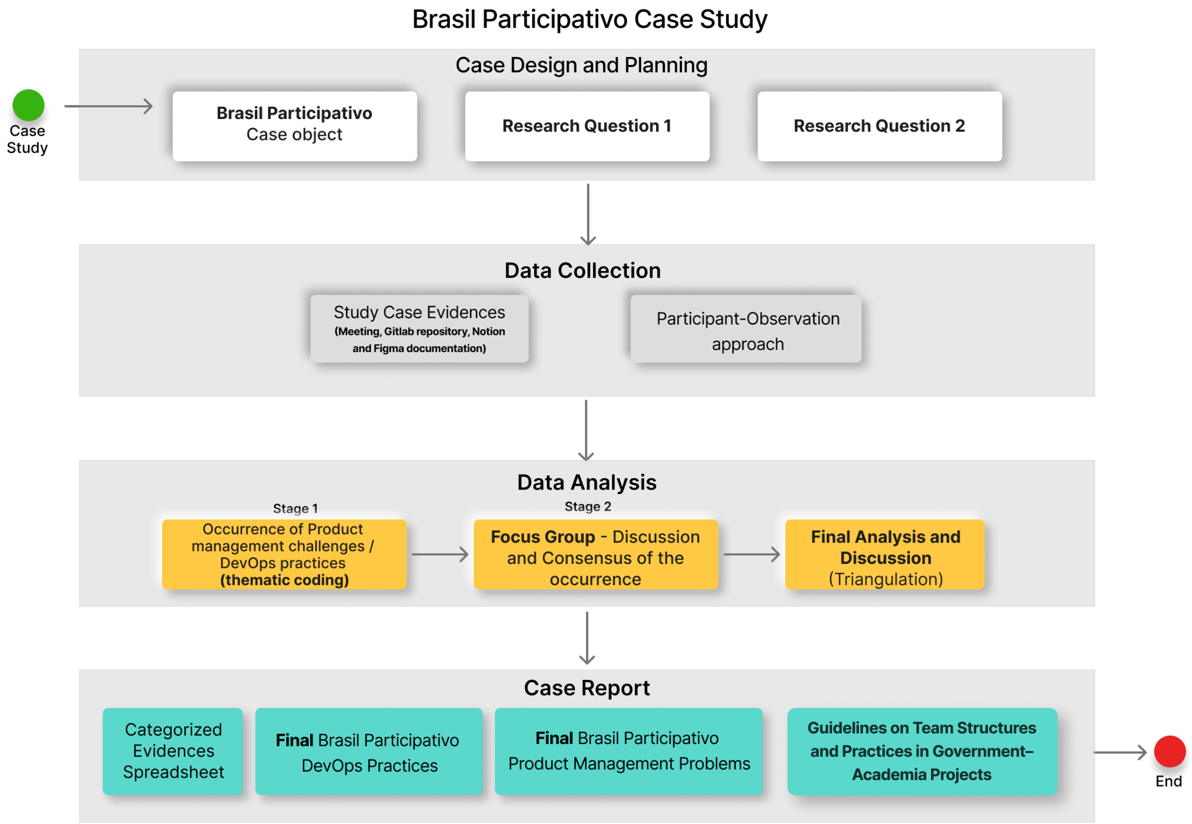
We collected evidence from the project, including meeting records, GitLab repositories, Notion project files, and Figma designs. With this, we identified three distinct phases in team composition and practices that guided the Data analysis step.

The Data analysis step will follow a two-stage approach, as illustrated in Figure 1.1, structured to address the proposed research questions. First, we analyzed the artifacts collected over 34 months of project activity using the participant-observation approach. To guide this analysis, we employed thematic coding; rather than constructing a codebook from scratch, we grounded our coding scheme in the product management challenges identified by Springer and Miler (2022) and the project management practices defined in the DevOps team taxonomy proposed by Díaz et al. (2024). We use the evidence to support the determination of the codes' identification. For each phase, we first describe the context and the decisions that drove changes in team structure and ways of working.

The second stage employs the focus group technique (Kontio et al., 2008) with key project stakeholders to refine and validate the findings from the initial analysis. This step enriched the discussion by capturing the perceptions of participants with different roles and vantage points within the project, strengthening the credibility of the conclusions drawn. Following focus group validation, we triangulate the findings across DevOps practices, team structures, and product management challenges to surface intersections and tensions between technical delivery optimization and product management effectiveness. From this synthesis, we derive a set of best practices and team configurations to address these challenges in the specific context of government-academia co-development projects.

For the report step, the research yields three primary contributions: a comprehensive characterization of a government-academia co-development project, including its phases and their relationship to product management challenges; a set of recommendations for structuring teams in government-academia collaborations oriented toward long-term product development; and a synthesis of key insights on team structures and product management in technopolitical software contexts.

Figure (1.1) The Case study steps followed for this study.



1.4 Claimed Contributions

This research provides four primary contributions to software engineering knowledge and practice:

C1. Empirical Evidence on the Relationship Between Team Structures and Product Management Effectiveness. Through qualitative case study analysis, we provide empirical evidence demonstrating how specific practices and team structure affect common software product management challenges such as requirement prioritization, stakeholder alignment, and delivery coordination. This evidence helps bridge the gap between organizational theory and practical implementation.

C2. Guidelines for characterizing and choosing Appropriate Team Structures and Practices in Government-Academia Collaborations. Based on our analysis of team structures and practices and their effectiveness in different contexts, we provide actionable guidelines for organizations engaging in government-academia software projects. These guidelines consider factors such as project complexity, stakeholder diversity, regulatory requirements, and resource constraints.

We published an article regarding the co-development process of Brasil Participativo and its multiple actors' roles in the launch of the platform Silva Rocha Aguiar et al. (2024), and we intended to publish the results of this research:

- Rocha, Carla; Alves, Isaque; Gomes, Leonardo; Pinos, Bruna; Bellix, Laila; Parra, Henrique. Colaboração multissetorial para desenvolvimento e manutenção de soluções tecnológicas de participação: o caso do Brasil Participativo. RSP, vol. 75, nº a, p. 89- 107, ago. 2024.

1.5 Thesis Structure

This thesis is organized into six chapters. Chapter 1 introduces the motivation for the study, outlines the research problem, presents the research questions, states the claimed contributions, and describes the overall structure of the work. Chapter 2 provides the theoretical background, reviewing the literature on software product management, Agile and DevOps team structures, government–academia co-development, and technopolitical platforms such as Decidim and Brasil Participativo.

Chapter 3 details the research design, justifying the adoption of a holistic case-study methodology and describing the data-collection procedures, including document analysis, participant observation, and focus groups, as well as the analytical techniques, including directed content analysis and coding.

The Chapter 4 introduces what the Brasil Participativo is, in terms of innovation obtained in this collaborative development project, and the Chapter 5 portrays the 3 phases of the Brasil Participativo project, mapping the consolidation of the researcher's and focus group's findings on product management challenges and observed team structures. In Section 5.6, we discuss the differences between the phases, the problems they solve, and align on best practices for managing product co-development to enable continuous delivery.

Finally, Chapter 6 concludes the thesis by summarizing the main results, emphasizing the original contributions, acknowledging the limitations, and pointing to promising directions for future research on product management, DevOps, and government-academia co-development collaboration.

Chapter 2

Background

In this chapter, we describe the key concepts of this research. It introduces the Continuous Delivery method and its relationship to DevOps, Software Product Management frameworks, and literature on the challenges of applying product management in projects. We dive deeper into Agile and DevOps practices and, finally, the concept of co-development in government settings and the introduction of technopolitical software such as Decidim and Brasil Participativo.

2.1 Continuous Delivery

Continuous Delivery serves as the central concept of this research. Its origins are rooted in the first principle of the Agile Manifesto, which asserts that the highest priority is to satisfy the customer through early and continuous delivery of valuable software (Agile Alliance, 2001). Consistent with this principle, Continuous Delivery is defined as a software engineering approach in which teams consistently produce valuable software in short cycles and ensure its reliable release at any time (Chen, 2015).

From a technical perspective, Continuous Delivery aims to minimize the time between software modification and deployment to production. By shortening this feedback loop, defects are detected earlier, and the risks associated with releases become more visible and manageable (Humble and Farley, 2010). This approach shifts software delivery from infrequent, high-risk events to a continuous and controlled process.

However, Continuous Delivery is not limited to technical practices; it also plays a role in product development. Successful product development depends on delivering the right products to the right markets at the right time (Ebert, 2014). Continuous Delivery allows organizations to build products that better address real user needs, since it has been associated with improved product quality, more reliable releases, and faster feedback cycles (Chen, 2015).

DevOps promotes practices such as continuous integration, continuous delivery, continuous deployment, automated testing, infrastructure-as-code, and automated releases to support frequent, reliable software delivery (Chen, 2015; Macarthy and Bass, 2020). The automation enabled by Continuous Delivery affects various software engineering activities, including development, testing, and deployment, and its influence extends beyond technical workflows. Because release activities involve multiple organizational units such as development, operations, and business, automation also shapes organizational structures and collaboration patterns, necessitating increased integration and coordination across teams (Leite et al., 2020).

As an example of a successful adoption of continuous delivery in a co-development setting, Siqueira et al. (2018) described the experience implementing it in the 30-month-long Brazilian Public Software portal (SPB) project. Overcoming the challenge of walking through a series of bureaucratic procedures to acess the government infrastruture and deploy features, and to remove the belief among government staff that any project in partnership with a university is doomed to fail, the pipeline implemented with automated acceptance tests and validation environment brought the result of 64% of the 84 versions released being in the 12 months period aftware its creation. Also, it increased transparency in the work and gained recognition from government staff for the quality of their deliveries.

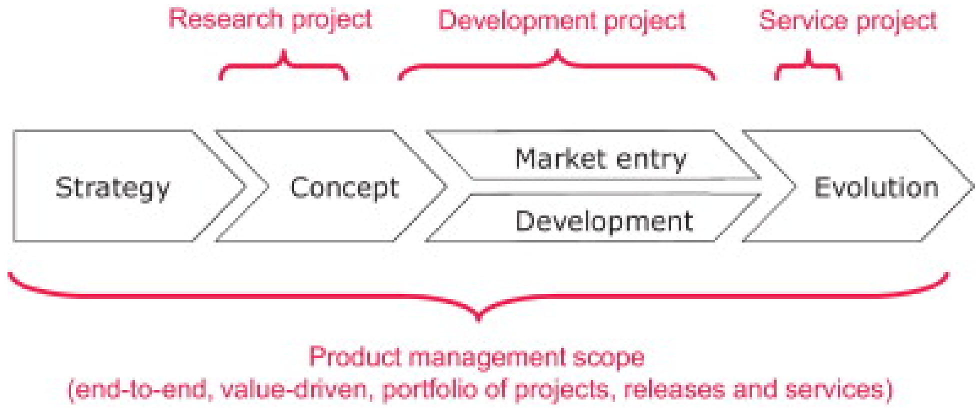
2.2 Product Management

Kittlaus and Fricker (2017) defines a product as a combination of goods and services that an organization develops to serve commercial objectives while transferring specific rights and value to customers. In the software context, a software product is characterized as having software as its primary component. For some software, the designation as a "product" arises not from the purchase transaction itself, but from the strategic intention to make it available to third parties as a valuable entity (Kittlaus and Fricker, 2017).

Product management is a discipline used across many industries to plan and coordinate the relevant aspects of a product to optimize product success. (Kittlaus and Fricker, 2017). It manages a product from its inception through market introduction, customer delivery, and service to maximize value generation for the business (Figure 2.1) (Ebert and Brinkkemper, 2014). From a software perspective, software product management is a set of processes that involve distinct responsibilities, coordinate activities across the company, and aim to define, introduce, develop, grow, maintain, and withdraw a software product from the market (Springer and Miler, 2022).

The distinction between project management and product management centers on their respective objectives. Project management involves overseeing a temporary en-

Figure (2.1) Product management in the entire product-life cycle (Ebert and Brinkkemper, 2014).



deavor to deliver a specific product within defined time, budget, and quality constraints. In contrast, product management focuses on the product's overall market success and ongoing development, including subsequent releases and associated services (Ebert and Brinkkemper, 2014).

The origins of product management are rooted in marketing, particularly in branding, communication, and pricing. However, with the rise of new industries such as technology, the focus of product management shifted toward the product itself, emphasizing how to define its value and how to develop software products effectively (Eriksson, 2011).

According to the ProdBOK, the product management lifecycle comprises seven phases: conceive, plan, develop, qualify, launch, deliver, and retire (Geracie and Eppinger, 2013). To guide these steps and ensure that market needs are effectively translated into organizational outcomes, the ProdBOK assigns this responsibility to the product manager. As the voice of the customer, the product manager is responsible for balancing and aligning expectations across multiple departments, overseeing product planning and articulating requirements. ProdBOK highlights that the development phase is the company's most resource-intensive stage, requiring substantial investment in design, implementation, and product testing.

The ISPMA framework adopts a holistic perspective on software product management, structuring it into phases that span the entire product lifecycle (Kittlaus and Fricker, 2017). The activities typically performed by a software product manager are part of the *Core SPM*, which includes *Product Strategy* (e.g., product definition, sourcing, and pricing) and *Product Planning* (e.g., roadmapping and product planning). However, *Orchestration*, where development activities are located, is considered outside the core domain. In practice, many organizations apply processes that only encompass the software development life cycle (SDLC) rather than the broader product development life cycle (PDLC). Agile methodologies are primarily applied in the former, and, in Scrum, they define a specific orchestration role, the *Product Owner*, whose task is to optimize collaboration among all units to achieve product-related goals (Kittlaus and Fricker, 2017; Schwaber and Sutherland, 2020).

Within the ISPMA framework, the development unit includes software engineers, user experience specialists, and quality assurance experts, while product management coordinates requirements engineering, often in collaboration with the product owner in agile contexts (Kittlaus and Fricker, 2017). Despite this structure, several recurring areas of conflict can be observed between software product management and development:

1. **Prioritization of requirements**: product managers tend to emphasize functional requirements, whereas development teams often advocate for prioritizing code quality improvements.

2. **Deviation from backlog**: developers may disregard previously defined requirements, arguing that delivery schedules are unrealistic.

3. **Implementation disputes**: product managers may disagree with how developers implement certain functionalities.

4. **Changing requirements**: adjustments driven by customer situations or market dynamics can generate friction in ongoing development efforts.

5. **Misalignment between strategy and cadence**: the rapid delivery cycle of agile sprints may clash with the long-term product strategy defined by product management.

6. **Deployment frequency**: disagreements arise when product managers prefer controlled release timing, while the development and operations team, enabled by DevOps practices, promote continuous delivery and rapid deployment (Kittlaus and Fricker, 2017).

(Springer and Miler, 2022), through a study that included a systematic literature review, interviews, and a survey, described common product management problems and their perceived frequency and severity. Unlike Maglyas et al. (2013), which listed 5 main problems, Springer and Miler (2022) summarized 27 possible problems. The list of problems is provided in 2.1 and encompasses various aspects, as Software Product Management encompasses many processes and responsibilities that connect different activities within the company. In a recent work (Springer et al., 2023), this problem was divided into 8 categories, and solutions were proposed for the 5 most common problems identified previously.

Table (2.1) Software Product Management Problems (Adapted from Springer and Miler (2022)).

ID	Problem Description
P74	Determining the true value of the product that the customer needs
P35	Strategy and priorities are changing frequently
P9	Technical debt
P64	Working in silos (problem with communication, synchronization between teams)
P69	Balancing between reactive and proactive work. When comparing hypotheses to facts, hypotheses lose value to facts (such as client requests and bugs). Managing requirements instead of identifying problems and opportunities, seeking innovation.
P26	Lack of support for research (no resources allocated to the team)
P85	Lack of automated testing
P82	Product Manager role not clearly defined and communicated in the organization (what the role is about, what the responsibilities and objectives are, decisiveness)
P48	Lack of user research
P51	Roadmap focused on features instead of goals and business value
P18	Lack of market research, no understanding of business and trends in the industry
P8	Different expectations about product management communication per stakeholder (high/low level)
P6	Lack of continuous integration and delivery (impact on 'time to market')
P25	Limited access to users in order to do research
P16	Lack of synchronization between product management, marketing, and sales units
P1	No company strategy
P44	Teams are not autonomous and self-organized (difficult to organize, dispersed both responsibility and decision-making)
P10	Lack of user analytics data
P19	Teams are not Agile, they just follow rules and do not use experimentation and a learning process
P13	The team's lack of motivation due to a lack of understanding of why they are doing things - no commitment to achieve goals
P41	High expectations from external partners, which cannot be met
P3	Wrong data analytics setup
P12	Low software quality
P78	Lack of skills to use and analyze the data
P43	Lack of trust in the product team (micro-management)
P27	Unqualified team members (individuals)
P56	Price management is always experimentation burdened with risk

2.2.1 Agile Practices

Agile emerged in 2001 with the publication of the Manifesto for Agile Software Development, which emphasized values such as individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan (Chow and Cao, 2008). Rather than being a single methodology, Agile is a broad umbrella of practices, values, and principles that share an iterative and incremental approach to software development (Alsaqqa et al., 2020). This makes Agile a great fit for situations where users' wants keep changing, since teams can easily adjust what they're building without redoing much work (Davis et al., 1988).

One of the main advantages of Agile over traditional approaches lies in its ability to handle change, particularly change arising from user requests or market dynamics, which are frequent sources of problems in sequential life-cycle models (Alsaqqa et al., 2020). Agile methods promote rapid prototyping, early delivery of intermediate versions, and close collaboration with users, enabling feedback loops (Guntamukkala et al., 2006). Furthermore, frameworks like Scrum reinforce product management by continuously improving the product, the team, and the working environment, while assigning the Product Owner role to maximize value delivery and maintain an effective backlog (Schwaber and Sutherland, 2020).

The success of Agile projects, however, depends on specific conditions. An empirical study by Chow and Cao (2008) identified 12 critical success factors, consolidated into five categories: organizational, people, process, technical, and project. These factors include delivery strategy, agile software engineering practices, team capability, effective project management processes, a supportive team environment, and strong customer involvement. They align success with four attributes: quality, scope, timeliness, and cost.

2.2.2 DevOps Practices

DevOps emerged as both a cultural and technical movement aimed at breaking down organizational silos, fostering empathetic collaboration, and enabling continuous development and delivery processes, particularly across development and IT operations (Díaz et al., 2021; Leite et al., 2019; Luz et al., 2019; Lwakatare et al., 2016; Smeds et al., 2015). Like Agile, DevOps emphasizes short cycles, rapid feedback, and strong customer involvement, but it expands these practices (Leite et al., 2019).

Its adoption has benefits such as increased organizational IT performance and productivity, reduced software lifecycle costs, improved operational efficiency and effectiveness, better software product quality, and greater alignment between development and operations teams (Luz et al., 2018).

One of the main challenges of adopting DevOps is its ambiguous definition, which is variously described as a culture, a role, or just a collection of tools. This lack of clarity contributes to resistance and misunderstandings within organizations and, at times, leads to DevOps being dismissed as a "buzzword" (Smeds et al., 2015).

Based on a literature review, Leite et al. (2019) proposed a DevOps conceptual map, with concepts organized into four major categories: *Process*, *People*, *Delivery*, and *Runtime*. According to the authors, the **Process** category encompasses business-related concepts (eg, reducing risk and cost, complying with regulations, and improving product quality and customer satisfaction). **People** covers skills and concepts related to the culture of collaboration, essential for breaking down silos. From an engineering perspective, **Delivery** provides the concepts necessary for Continuous Delivery, while **Runtime** synthesizes the concepts required to guarantee the stability and reliability of services in a continuous delivery environment (Leite et al., 2019).

It shows that DevOps is more than tools and encompasses artifacts, roles, and team structures. Understanding this in the context of DevOps adoption is essential for developing successful implementation strategies. Therefore, grasping existing team taxonomy theories is critical for guiding companies toward more systematic DevOps adoption (Díaz et al., 2024).

Multiple studies have explored team structures and their impact on software delivery performance (Díaz et al., 2021; Leite et al., 2021; Luz et al., 2018, 2019; Smeds et al., 2015).

The study by Luz et al. (2018) presents a grounded theory analysis of successful DevOps adoption based on interviews with 15 practitioners across five countries, identifying "collaborative culture" as the core category for DevOps implementation.

The authors define collaborative culture as eliminating silos between development and operations teams. Exercising activities such as deployment, monitoring, and day-to-day activities, and trusting and empowering the development team to do these activities can help reduce conflicts between teams.

This collaborative approach also requires "product thinking". Rather than isolated development or operations perspectives, the development team needs to consider that the product life continues after code is made available, and, on the operational side, the process should start before receiving this code for deployment. The study mentions that straightforward communication and shared responsibility are important for DevOps adoption, in contrast to traditional ticket-based systems and blame-oriented cultures, where the focus is on finding guilty parties instead of solving problems as a team.

The research cites another two categories of the adoption of DevOps: the enablers and the outcomes.

The enablers include automation (deployment, infrastructure provisioning, testing), since the manual task can cause silos because it is commonly concentrated in a single person or team for execution; knowledge sharing (of activities, processes, and procedures) to all team members to gain a better insight about the software production; and transparency in execution and decision making.

The expected outcomes of successful DevOps adoption include agility through continuous integration and resilience of the application through auto-scaling and recovery automation.

The authors provide a model for DevOps adoption and applied the three-step model at the Brazilian Federal Court of Accounts, showing benefits including increased deployment frequency (from weekly to up to 29 deployments in a day) and reduced downtime, corroborating their theory that focusing on collaborative culture rather than tools alone leads to more sustainable DevOps adoption.

Leite et al. (2021) grounded theory study of 37 IT professionals across several organizations provides evidence to understand how software teams organize to achieve continuous delivery goals. Their research identified four common organizational structures: siloed departments, classical DevOps, cross-functional teams, and platform teams.

In Siloed departments, there is a strict separation between development and operations, where each department is guided by its own interests, developers have limited awareness of what is happening in production, neglect non-functional requirements, and implement minimal DevOps initiatives. This could lead organizations to achieve lower delivery performance.

Classical DevOps structures preserve separate teams, whereas they emphasize enhanced collaboration and shared responsibilities, particularly around non-functional requirements and incident response. In this structure, the alignment with departments is important, but this is not trivial to achieve.

Cross-functional teams embody the "you build it, you run it" philosophy, where single teams handle both software development and infrastructure management, providing autonomy. This structure could potentially struggle with skill gaps and a lack of specialized infrastructure knowledge.

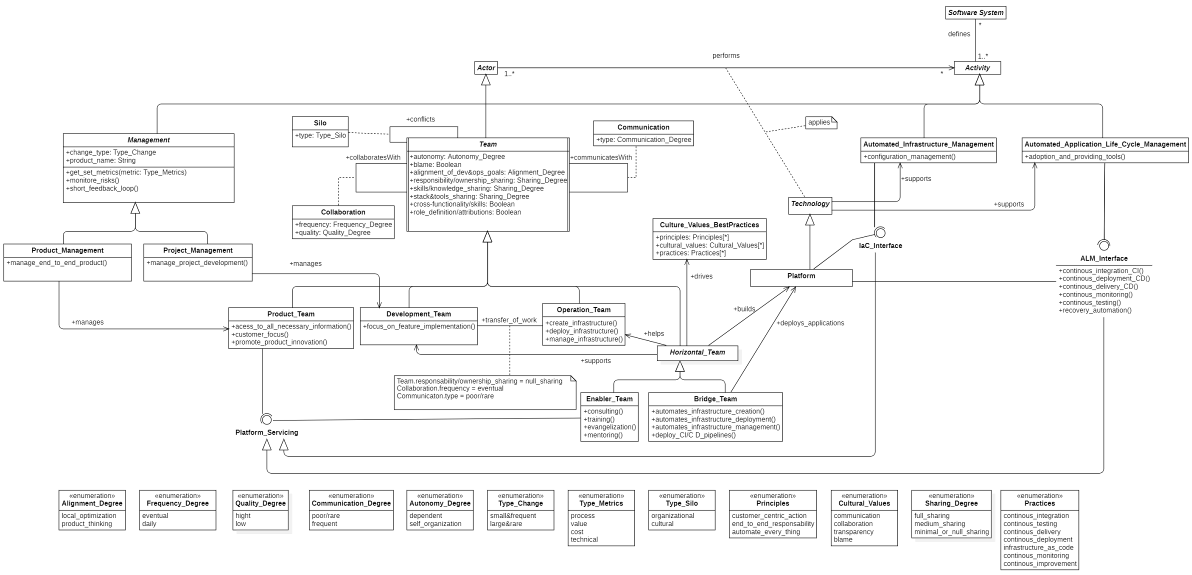
Platform teams represent the most advanced structure, in which dedicated infrastructure teams build automated, self-service platforms that enable development teams to deploy and operate their applications independently while abstracting infrastructure complexity.

The identification of transitional patterns between structures indicates that organizations typically progress through these models sequentially, with platform teams representing the culmination of DevOps organizational maturity rather than an immediate transformation target for traditional siloed environments.

To unite the concepts proposed by researchers regarding DevOps adoption and team structures that support these practices, Díaz et al. (2024) conducted a grounded theory study to harmonize DevOps taxonomies. Their work resulted in a systematic codebook that organizes DevOps practices into categories such as collaboration, automation, monitoring, and sharing. For example, the codebook lists practices like infrastructure-as-code, involving operations staff in development stand-ups, and continuous measurement of software metrics to guide decision making. This effort provides much-needed standardization, helping organizations adopt DevOps practices more consistently.

After defining the DevOps taxonomies, Díaz et al. (2024) mapped the relationship of team structures and these taxonomies, defining then the constructs and the propositions of the theory. For instance, the construct *Product_Team* is a construct responsible for a product/service, thus it is common to have the construct mapped in the codebook *responsibility/ownership_sharing* with different levels (*medium_sharing or full_sharing, i.e.*). The full map of the constructs and propositions is represented in Figure 2.2.

Figure (2.2) Full representation of the DevOps Constructs and Proposition from Díaz et al. (2024) theory



2.3 Co-development and Government software

Electronic government (e-government) is understood as the application of information and communication technologies (ICT) to support government operations, deliver services, and engage citizens in decision-making. It is not merely a digitalization of existing processes but rather a special case of ICT-enabled business process change that requires a holistic view of organizations, their cultures, systems, and stakeholders (Anthopoulos et al., 2016; Scholl, 2003). Despite its potential to improve efficiency, transparency, and participation, e-government projects face high rates of partial or total failure due to complexity, political influences, and design–reality gaps (Anthopoulos et al., 2016; Goldfinch, 2007).

To overcome these challenges, governments increasingly adopt open and collaborative approaches. The use of Free/Libre/Open Source Software (FLOSS) facilitates co-development, enabling both public and external actors to contribute to innovation while reducing dependence on private vendors. FLOSS adoption also favors transparency and provides citizens with easier access to services and data, since source code and development processes are openly available. Moreover, the FLOSS model is inherently collaborative, bringing together diverse contributors who can use, study, modify, and redistribute software while fostering participation and shared innovation (Kon et al., 2011).

FLOSS adoption favors transparency and facilitates citizen access to data and services, while its open development model promotes participation and shared innovation. FLOSS practices, such as open communication, modular design, and community engagement, align with agile values and enhance cooperation across institutions. FLOSS contributes to public-sector governance by promoting transparency, facilitating adaptability, and providing mechanisms that enhance the sustainability of software initiatives. (Kon et al., 2011; Mergel, 2015; Wen et al., 2018).

When focus in constructing a new product, the government-academia collaboration follows under the definition of collaborative product development (CPD), which can be defined as two or more partners joining resources and experiences to design or develop a new or improved product (Büyüközkan and Arsenyan, 2012).

Collaborative software development in the public sector benefits from partnerships between government, academia, and industry. Academia introduces cutting-edge methodologies, while the government provides real-world challenges and resources. However, such collaborations face cultural and managerial mismatches that need to be harmonized. Studies highlight that adopting FLOSS and agile practices increases the chances of success by aligning diverse workflows and promoting a shared understanding of goals (Wen et al., 2020).

Concrete examples in Brazil illustrate how FLOSS-based initiatives can be implemented through government–academia collaboration. The Portal do Software Público (SPB) was restructured through a partnership between the Ministry of Planning, the University of Brasília (UnB), and the University of São Paulo (USP) (Meirelles et al., 2017; Siqueira et al., 2018; Wen et al., 2018, 2020). Wen et al. (2018) highlights that the success of this initiative was supported by factors such as the alignment of goals between government and academia, the use of FLOSS ecosystems to ensure openness, and the adoption of collaborative infrastructures (mailing lists, issue trackers, and social networking features) that fostered communication across institutional boundaries. These conditions enabled a sustainable co-development environment and reduced the asymmetry between government demands and academic practices (Wen et al., 2018, 2020).

Similarly, the Brasil Participativo platform was structured as a multisectoral collaboration that involved the federal government, universities, and industry partners. Silva Rocha Aguiar et al. (2024) emphasizes that this project formalized a governance structure in which roles were distributed among the actors: the federal government provided institutional legitimacy and policy direction, academia contributed technical expertise and research capabilities, and industry partners ensured technological development and operational support. This configuration illustrates how FLOSS infrastructures, combined with structured partnerships, can sustain large-scale digital participation platforms (Silva Rocha Aguiar et al., 2024).

2.4 Technopolitics and Technopolitical Softwares

The Internet has often been celebrated as a force capable of enhancing democracy, yet after more than three decades, it has largely mirrored existing social and political structures rather than transforming them (Kurban et al., 2017). Even so, the spread of information and communication technologies (ICTs), including software, web platforms, and mobile applications, has accelerated and facilitated processes of idea collection, public discussion, decision-making, voting, and content creation, expanding opportunities for civic participation (Kurban et al., 2017).

Within this context, technopolitics emerges as a key concept to describe how technology and politics intersect. It refers to the ways in which political actors appropriate and mobilize technical artefacts to pursue their objectives, whether through centralized top-down approaches where information remains controlled by institutions, or through distributed bottom-up practices where information is co-produced and shared across networks, fostering more democratic and participatory forms of governance (Kurban et al., 2017).

Kellner (2001) conceives technopolitics as a strategic means of empowering citizens, emphasizing that despite some resistance to the use of ICTs, these technologies are vital to the democratic project because they create new arenas of political struggle for groups excluded from mainstream media, thereby enhancing possibilities for resistance and intervention. He regards technology as an autonomous agent that can be appropriated by opposing actors for divergent political purposes. His perspective carries a clear normative orientation, arguing that the appropriation of ICTs should serve democratic aims, with technopolitics understood not as an end in itself but as an instrument of struggle in support of democratic revolutions.

As Kurban et al. (2017) emphasizes, technopolitics reshapes the mediations between people and institutions. Interactions with technological affordances and constraints produce new strategies that reconfigure power relations. In the short term, this process alters practices and routines, while in the long term, it transforms organizations and norms. The potential outcomes include more open and distributed forms of governance, but such outcomes require infrastructures that embed participation at their core rather than treating citizens as passive recipients of services.

2.4.1 Brasil Participativo

In the Brazilian context, the Constitution values social participation not only as part of state control but also as a key component of the decision-making process for public policies and their implementation (da Silva et al., 2005).

The National Social Participation Policy (PNPS) is a set of concepts and guidelines that aim to consolidate the adoption of mechanisms for this type of participation, enabling the sharing of decisions between the federal government and civil society. This policy outlines the methodologies for participation and the incorporation of multiple means of participation, utilizing information and communication technologies (ICTs), with a focus on open-source software and applications (Secretaria-Geral da Presidência da República, 2014).

As a methodology for participation in Brazilian politics, there are the National Conferences, which are instances of debates with representatives of civil society and government, which begin at the municipal and state levels and reach the national level, where specific themes are formulated and evaluated (such as health, education, and youth) (Secretaria-Geral da Presidência da República, 2014). Another form of participation is public consultation, a virtual process that collects written opinions from the population regarding a specific document or subject (Governo Federal do Brasil, 2025).

There is also the national plan, created by law and intended to define principles, guidelines, objectives, and goals to guide the development and implementation of policies to address specific topics (Ministério da Cultura, 2025).

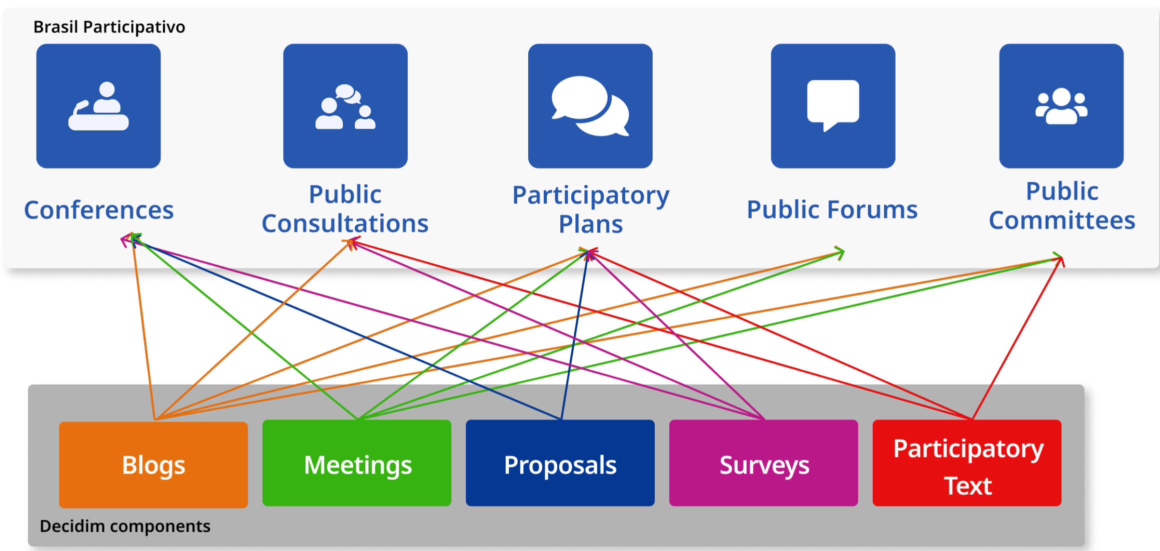
Brasil Participativo is a digital platform designed to strengthen social participation and integrate all the methodologies of participation mentioned. It was developed as part of a collaborative project between the University of Brasília (UnB) and the Office of the Presidency of the Republic. An early decision was to adopt the Decidim framework, as it enables flexible configuration of participation through its spaces and components. Moreover, being an open-source platform, it ensures opportunities for community contributions and long-term sustainability. The goal of Brasil Participativo is to increase citizen participation and make decision-making processes more accessible and transparent.

The platform's first participatory process was the 2024–2027 Participatory Pluriannual Plan (PPA), held between May and July 2023. Decidim, the open-source software underlying the solution, was rapidly adapted to Brazil's specific requirements, with a focus on functionality and usability to meet those demands.

During the implementation of the PPA, an opportunity was identified to further improve the Decidim-based platform to encompass all participatory processes in the Brazilian context. In addition, the goal was to bring these participatory experiences together in a single environment by migrating the existing public consultation platform at the time, **Participa+ Brasil**[1]. The project then evolved into building and adapting functionalities to accommodate the diverse methodologies of participation in Brazil, ensuring that online forms of participation could continue to take place successfully.

Figure 2.3 illustrates the 5 types of participation processes within the Brasil Participativo and the Decidim components used to build them on the platform. In Subsection 2.4.2, there is a further description of this component and an overview of the Decidim platform architecture.

Figure (2.3) 5 types of participation processes in Brazil Participation and Decidim components for building the processes



[1]https://www.gov.br/participamaisbrasil/

2.4.2 Decidim

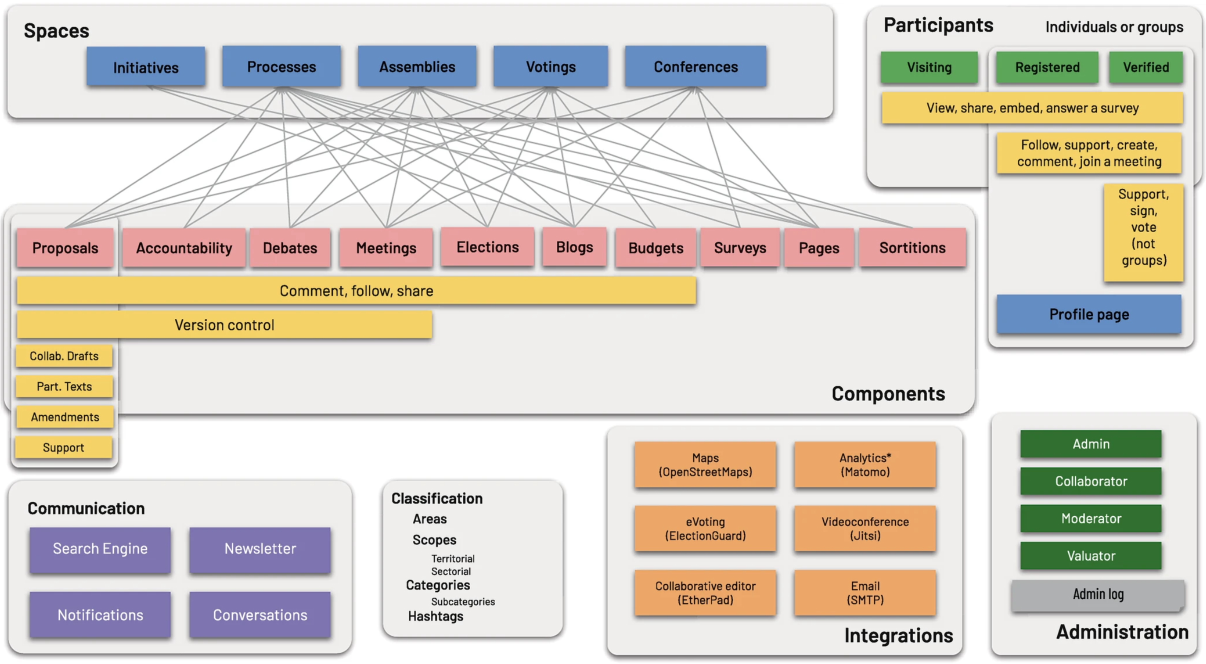
Decidim, as presented by Barandiaran et al. (2024), exemplifies what technopolitical software can be. It is a "public-common, free, and open digital infrastructure for participatory democracy", built collaboratively in Ruby on Rails (Barandiaran et al., 2024).

Decidim operates simultaneously on political, technopolitical, and technical planes. Politically, it shapes the kinds of democratic processes and decisions institutions can host. Technopolitically, it embeds politics in its design principles, data policies, and community governance. It is primarily a reflexive, decision-making, and operative space on how technologies structure political processes and vice versa. Technically, it encompasses software code, licensing, and operational workflows Barandiaran et al. (2024).

Barandiaran et al. (2024) explains the use of Decidim as follows: Users interact through mechanisms known as **components** within different **spaces** that guide and organize their actions towards specific results. **Participatory spaces** are the frameworks that establish how participation occurs and serve as the channels through which members of an organisation engage with the platform. These spaces include **Assemblies**, **Processes**, **Initiatives**, **Votings** and **Conferences**. Within these spaces, participants make use of **components**, which provide concrete participatory mechanisms such as meetings, proposals, blogs, debates, static information pages, surveys, results, and comments. The combination of spaces and components gives the platform its flexibility to articulate discussions, structure decision-making, and record outcomes.

The representation of Decidim's components and architecture, including the integration of components and participatory spaces, is shown in Figure 2.4.

Figure (2.4) Decidim's functional architecture, showing a combination of components in participatory spaces (Barandiaran et al., 2024).



Chapter 3

Research Design

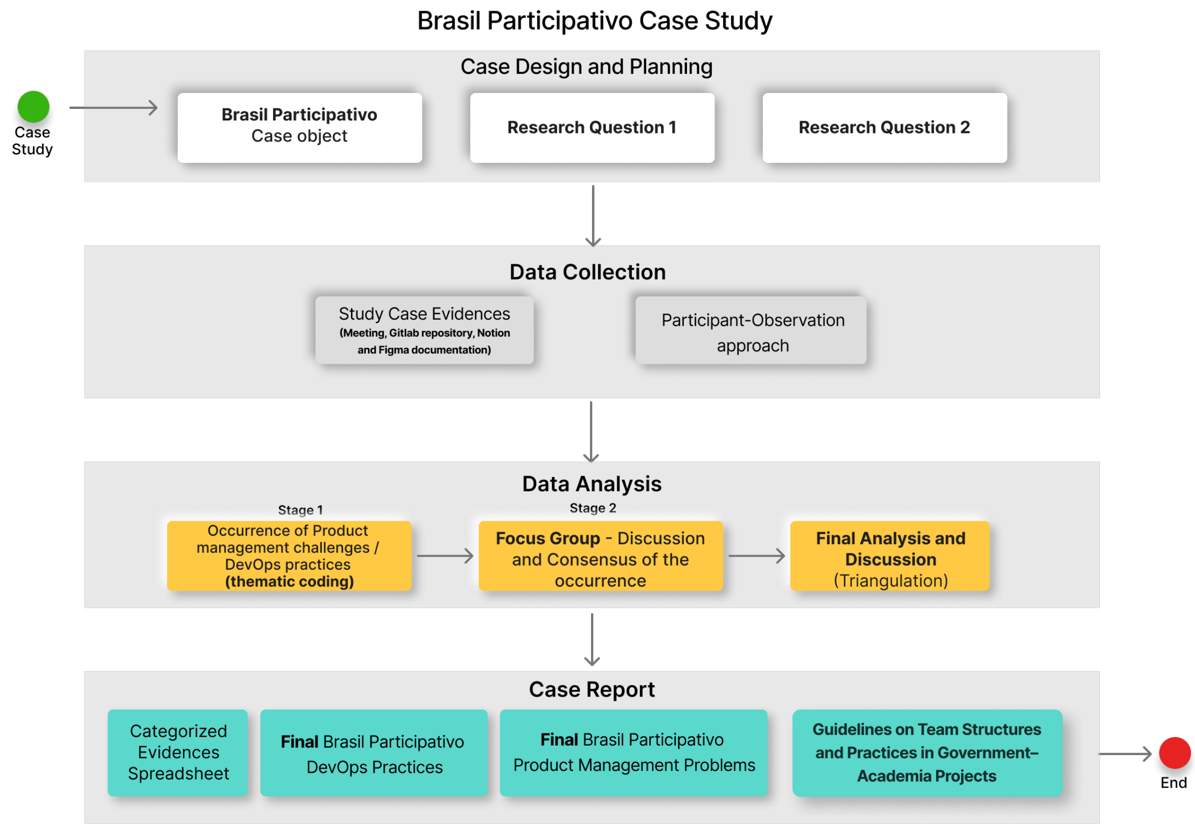
This thesis aims to highlight the product management challenges in e-government software development within this type of partnership and to examine how agile practices, such as continuous delivery and team configuration, affect the success of an e-government product. For that, the research proposes to answer the following questions:

RQ1. How do different continuous delivery practices and team configurations affect the product management of a software in a government-academia collaboration context?

RQ2. What practices and team structures favor the development of a government-academia collaborative product?

This chapter details the case study protocol adopted in this research, grounded in the methodological guidelines proposed by Yin (2018) and Wohlin et al. (2012). It presents the overall study design (Figure 3.1), including the planning procedures and the development of the case study protocol. Then, it describes the data collection strategy, using case evidence and participant observation, highlighting the researcher's active role within the project context. In this chapter, the data analysis approach is explained, based on coding and thematic analysis supported by an established codebook. Finally, it discusses how the study's validity is addressed through the use of focus groups, following the recommendations of Kontio et al. (2008), and the triangulation of multiple sources of evidence, ensuring the reliability and credibility of the findings.

Figure (3.1) The Case study steps followed for this study. (show also in Section 1.3)



3.1 Case Study

The Brasil Participativo project is a complex collaboration among three distinct sectors: government, academia, and industry. Each sector offers specific expertise, workflows, and cultural practices, creating a multifaceted environment in which team dynamics are intrinsically linked to their operational contexts. As a software platform operating on a national scale with diverse user profiles, it creates a non-deterministic environment that cannot be replicated under controlled conditions.

Given these characteristics, the case study emerges as the most appropriate research strategy. As noted in Yin (2018), this empirical method is designed to investigate contemporary phenomena in their real-world contexts, particularly when the boundaries between the phenomenon and its context are not clearly evident. In software engineering, this approach is particularly preferred when the subjects of study cannot be studied in isolation, thus favoring realism over control (Runeson and Höst, 2009; Wohlin et al., 2012).

This research is classified as holistic, since it regards only one object of study, the Brasil Participativo product, and it can also be distinguished for its purpose as an exploratory case study, since it seeks to understand what is happening, seeking new insights and generating ideas and hypotheses (Wohlin et al., 2012; Yin, 2014).

As described in Section 1.3, we examine the Brasil Participativo project through its three structural phases. For each phase, we analyze in greater depth the DevOps practices and product management problems encountered. The study design is illustrated in Figure 3.1.

There are common steps for conducting a case study (Wohlin et al., 2012; Yin, 2018). First, prepare and define the objectives and research questions for the case study (present in Section 1.3). Second, prepare the data collection procedures and protocol. Third, execute the study, collect the necessary data, then analyze the collected data, and finally report the findings. In the next sections, we explain the steps taken for this study in more detail.

The case study protocol is continually updated and serves as a guide for data collection, informs feedback from other researchers, and helps reduce the risk of missing relevant data sources (Runeson and Höst, 2009). For this specific research, we adapted the protocol proposed by Brereton et al. (2008) and created the complete case study protocol presented in Table 3.1.

Table (3.1) Present Case Study Protocol

Section	**Content**
Background	RQ1 and RQ2 describe in Section 1.2; Silva Rocha Aguiar et al. (2024), Wen et al. (2018, 2020), Springer and Miler (2022); Springer et al. (2023), Díaz et al. (2024)
Design	Single case, holistic design; **object of study**: Brasil Participativo; propositions derived from **research questions**
Selection	Current project which the researcher is working on; Opportunity to contribute to the literature regarding government-academia software product developing
Procedures and roles	The researcher acted as a member of the software product development team of the object of study
Data Collection	Participant-Observation, Project's open GitLab repository, Notion page, private Figma repository, team meetings, and project documentation.
Analysis	Based on problems and strategies described by Springer and Miler (2022) and the codes presented by Díaz et al. (2024), the researcher will associate evidence found in the case observed for each phase of the project
Plan validity	External researchers' validation of each phase outcome, and Focus Group
Study limitations:	This study is subject to researcher bias, as the author was actively involved in the Brazil Participativo project, which may have influenced the interpretation of the findings.

3.1.1 Data Collection

Data collection drew on two main sources of evidence: case evidence and participant observation.

Case Evidence

Data collection began with the organization of a case evidence inventory comprising technical artifacts and working documents produced throughout the project, including issues, commits, personal notes, project management documents, workflows, meeting notes, and presentations. Each item was logged with a unique identifier, source type, actual or approximate date, and its relationship to the case. This inventory also served to classify case materials into two groups: original historical materials, corresponding to artifacts produced at the time the activities took place; and retrospectively reconstructed records, corresponding to materials elaborated after the fact based on notes, memory, or the reorganization of existing artifacts.

As historical logs from earlier project phases were not fully available, the retrospective reconstruction of the case drew on archived materials. A case timeline was then constructed to locate relevant events, decisions, and transitions, and to guide the articulation between materials, interviews, and re-executions.

Participant Observation

In this work, in addition to the project documents, information was collected through participant observation, following principles endorsed by Yin (2018), such as the use of multiple sources of evidence and the maintenance of a chain of evidence.

Participant observation is a method of observation in which the researcher is not merely a passive observer. Frequently used in anthropological studies of different cultural or social groups, it can also be used in everyday settings, such as in large organizations (Yin, 2018).

This technique provides opportunities to access private events or documents, and the most distinctive opportunity of participant observation is the ability to perceive reality from someone's viewpoint (e.g., (Yin, 2018)). From a software development perspective, observing this process is limited, since "much of software development work takes place inside a person's head" (Seaman, 2008). The researcher, who serves as a software developer, can benefit from the study, as some software developers reveal their thought processes most naturally when communicating with other software developers (Seaman, 2008).

Challenges related to participant observation concern the potential biases introduced. The researcher may lose neutrality, align with the group studied, or prioritize participation over observation. It can also limit note-taking, questioning, and timely access to relevant events (Yin, 2018).

3.1.2 Data Analysis

Qualitative data refers to information expressed through words or images, whereas quantitative data is conveyed through numbers or discrete categories. Thus, the distinction lies in the form of representation rather than in the degree of subjectivity or objectivity (Seaman, 1999).

In Edmonds and Kennedy (2017), they propose four steps for analyzing qualitative research data. Those are:

- Prepare and organize the data for analysis: transcribing interviews, organizing notes, and getting survey answers
- Use memos for preliminary analysis: get notes and perceptions of the researcher's point of view
- Reduce data into themes through coding: coding is the process of exploring the data for themes, ideas, and categorization to make comparisons and to identify nan patterns that require further investigation
- Present the data in figures, tables, and narrative

Memoing is generally the technique used by qualitative researchers to help make conceptual links between raw data and abstractions, allowing them to explain the phenomena in its context. It records research ideas about concepts and their relationships within the data, thereby increasing the credibility of the findings (Edmonds and Kennedy, 2017).

Coding is an analytical technique used to systematically identify, analyze, and interpret patterns of meaning within a dataset, resulting in a statement or proposition that insightfully describes a phenomenon (Seaman, 2008). One analytical method typically used for systematic thematic coding (Ahmed et al., 2025; Braun and Clarke, 2006), when the goal is not to generate theory but to surface and interpret recurring patterns across the case evidence. The steps of Thematic analysis are (1) familiarization with data, (2) generating initial codes, (3) searching for themes, (4) reviewing themes, and (5) defining and naming themes (Braun and Clarke, 2006).

Familiarization with the data occurs organically during the collection process itself. The subsequent step, generating initial codes, aims to systematically identify meaningful segments within the data and associate them with relevant categories or themes.

Themes are not merely summaries of codes, but interpretive constructs that capture something meaningful about the data in relation to the research question (Braun and Clarke, 2006). By the end of the theme review step, the process yields a coding scheme, a codebook of themes and codes grounded in the data.

Rather than beginning with an empty codebook, we anchored our coding process in the pre-existing schemes drawn from Springer and Miler (2022) and Díaz et al. (2024), thereby adopting a directed content analysis approach, which is the approach where the researched uses existing theory or code to start and refine the code scheme, which generally is more structured than inductive alternatives (Hsieh and Shannon, 2005).

In this study, we adopt this codebook as our primary coding scheme to characterize themes related to product management and software delivery within the case evidence. We are interested in using the codebook to analyze the occurrence of each term in the project. The themes for which no supporting evidence was found in the collected data are removed. The final version of the codebook created by Díaz et al. (2024), and used as base for this study, is available in **Appendix-A**(see Section 7.1).

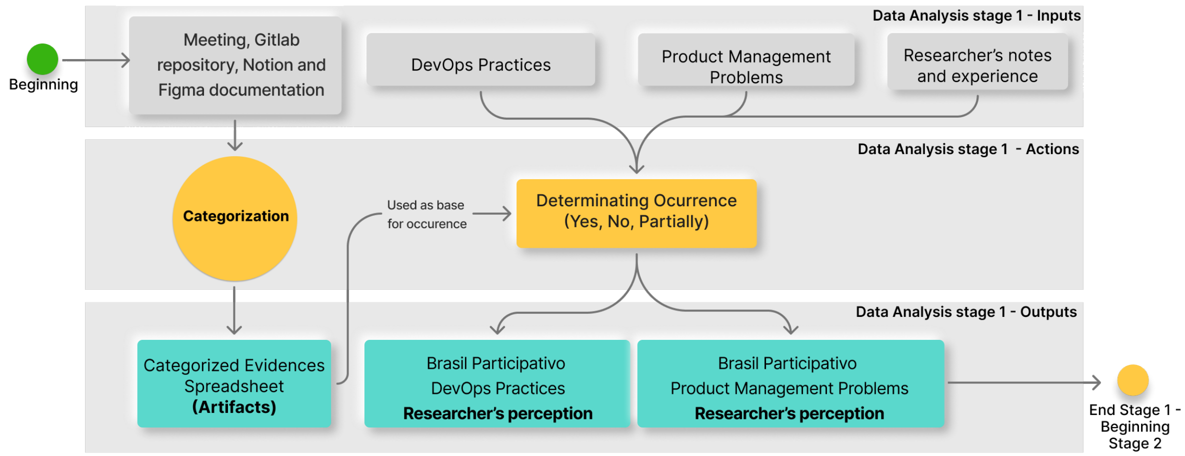
Díaz et al. (2024) proposed a harmonized DevOps taxonomy as a means of consolidating fragmented knowledge in the field. A primary output of the present work was a comprehensive **codebook** cataloguing DevOps practices alongside their descriptions and characterizations across different team structures, and mapping how each practice influences continuous delivery performance.

Additionally, from a product management perspective, we draw on the work of Springer and Miler (2022). Beyond a systematic review of the challenges reported in the field, their research resulted in a consolidated list of 27 software product management problems, categorized by frequency and severity (adapted in Table 2.1).

Since the coding scheme was established prior to analysis, the first stage of this research's data analysis is to observe the occurrence of these codes in the case study. As shown in Figure 3.2, the first stage is conducted by the study researcher, who evaluates the different levels of occurrence and adds artifacts (collected during the data collection step) to corroborate the evaluation.

The second step is to validate the analysis conducted by the focus group. The focus group technique will be applied to gather broader insights from other project members, enabling a structured refinement of the findings on team structures and the challenges faced within the project. Quotations collected in the focus group will be used to corroborate the final results of code occurrence.

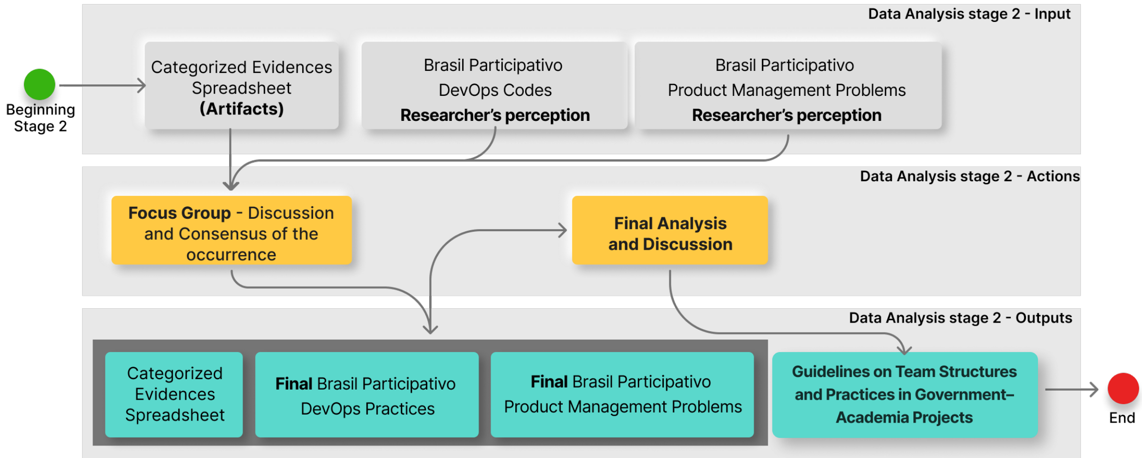
Figure (3.2) Data Analysis Stage 1 - Method to determine occurrence of the DevOps Codes and Product Management Problem in each Phase of the project (see Figure 3.1)



3.1.3 Plan Validity

Based on data analysis stage 1, we characterize the case study with respect to the practices adopted by the teams and the problems and solutions they encountered in managing both the project and the product. These preliminary findings are then validated and refined in stage 2 of the data analysis, as shown in Figure 3.3, through a focus group conducted in collaboration with key strategic stakeholders. After validation, the output is the final set of DevOps Practices and Product management problems identified, along with the team structures and artifacts used throughout the project, which provide input for the recommended compositions for a government-academia project.

Figure (3.3) Data Analysis Stage 2 - Inputs of Focus Group sessions and Results (see Figure 3.1)



A focus group is a qualitative research method that emerged in the 1950s in the social sciences and is designed to elicit personal perceptions from group members within a defined area of research interest (Kontio et al., 2008). Groups of typically 3 to 12 participants are conducted by a moderator-researcher who follows a predefined questioning structure and discusses a specific theme. This methodology offers low-cost benefits and provides candid, sometimes insightful information, but, as with some qualitative methods, it has the downside of group dynamics biases (Kontio et al., 2008).

Kontio et al. (2004) and Kontio et al. (2008) summarize the steps to conduct a focus group as:

- **Planning the focus group event**: defining the schedule and structure of the event. It usually takes two to three hours, so the number of issues to be covered has to be limited.
- **Selecting the participants**: It is important to recruit representative, insightful, and motivated participants, and depending on the type of research question, participants can vary in experience in the topic covered.
- **Conducting the focus group session**: The session should begin with an introduction in which the goals and ground rules are explained to participants. The moderator should facilitate the discussion without allowing their own opinions to influence it.
- **Analyzing the data and reporting the results**: the data analysis can use the methods used in qualitative data analysis.

For this study, the group discussion focused on discussing the product management challenges, solutions, and DevOps structure of Brasil Participativo, as first observed by the researcher. The debate generated in the focus group session is used to validate and refine the findings, helping answer **RQ2**: which team structures, practices, and artifacts are best for projects in the government-academia context.

The protocol of the focus group for this research:

- For each **Project Phase**:
  1. Invite active team members of the project;
  2. Present, in a meeting session, the first evaluation of the occurrence of the codes;
  3. Open for the participants to comment on the evaluation, adding their perspectives and additional evidence
  4. Ask for the consensus of the group regarding their evaluation
  5. Collect their evaluation and update the final codebook evaluation document.

Following the research design illustrated in Figure 3.1, the subsequent chapters describe the organizational structures identified in each phase of the Brasil Participativo project, as well as the results of the data collection and evidence analysis. Finally, we present the derived best practices for team structuring.

Chapter 4

Brasil Participativo

The collaborative project of the Brasil Participativo platform (Figure 4.4) has innovated not only in the technology itself, but also in the social participation methodology. In 2023, from March to May, the partnership designed a new participation methodology for the government's largest planning instruments, the 2024–2027 Participatory Pluriannual Plan (PPA), which was the first participatory process with a broad scope and encompassed all ministries' portfolios. Meanwhile, it also had to find and develop the right technology to handle this and future federal participation processes (Silva Rocha Aguiar et al., 2024).

In this first moment, we had different partnerships to launch the process in less than 3 months. The Social Participation Directorate of the General Secretariat of the Presidency of the Republic, along with other ministry portfolios and the external partner Instituto Cidade Democratica, and its participation specialist, articulated and developed the participation methodology. At the same time, the LabLivre - UnB investigated technologies capable of handling the predicted level of participation, being flexible enough to accommodate different participation levels, and including a community that could help with initial adoption, with the laborcatory also helping in the future. The project opted to use Decidim[1], a Ruby on Rails framework, which, at the time, was already adopted by different countries such as Italy, the United States, and Spain, and also Brazilian cities such as Contagem-MG and Belém-PA (Silva Rocha Aguiar et al., 2024).

Together, the Directorate and the LabLivre had also collaborated with Dataprev, a public company associated with the Ministry of Management and Innovation in Public Services (MGI), to provide technological infrastructure and, initially, developers, since the budget for the Brasil Participativo project was limited at the time. The different actions and responsibilities of the partners in this multisectoral arrangement are shown in Figure 4.1.

[1] https://decidim.org (Accessed on: 9 of April 2026)

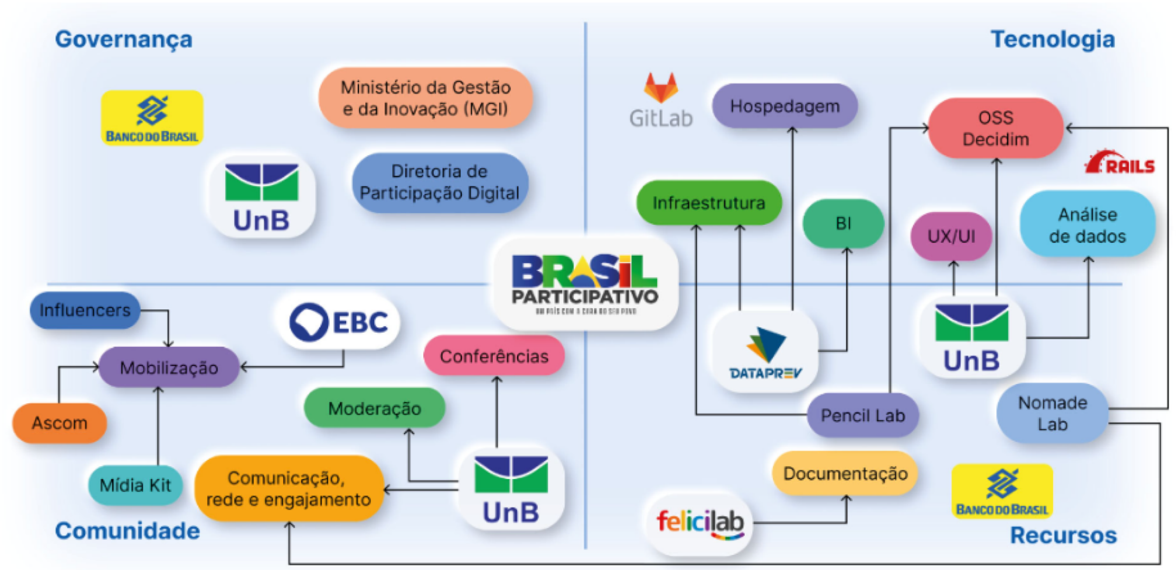


Figure (4.1) Composition of the multissectorial arrangement of the Brasil Participativo project (Silva Rocha Aguiar et al., 2024).

The launch of PPA and Brasil Participativo was an undeniable success. 2 months after its launch, the process harvested more than **1 million and 400 thousand** votes across **eight thousand** proposals, consolidating the partnership and demonstrating the innovation created in this collaborative development arrangement (Silva Rocha Aguiar et al., 2024). We restructured the Decidim platform to remove unnecessary functionalities, improved the design to be mobile-first after noticing during the process that approximately 90% of Brazilian users were more active using smartphones, simplified the participation, allowing users to vote with only two clicks, and used a strategy to promote the participatory process and the platform using digital influencers to arouse curiosity in the public.

These initial experiences influenced the platform's development and motivated innovations in participation methodologies. We can highlight 2 other processes: the Digital Stage of the 4th National Youth Conference and the National Culture Plan. Both processes directly affected the platform's requirements, driving user engagement and advances in digital social participation.

For the 4th National Youth Conference, we had to adapt Brasil Participativo to the participatory methodology used in the previous Youth Conference, namely, digital elections for proposals and delegates. National conferences in Brazil are traditionally conducted offline, with participants organizing municipal and state-level meetings to deliberate on proposals and elect delegates to represent them nationally. We created a gamified approach in which votes and proposal creation awarded users points, and, based on criteria and rankings, this citizen could be elected a delegate and participate in the national stage of the Conference. Along with the Youth Conference committee, we also helped develop the election criteria to ensure the representation of different population groups and diverse participation.

The technical innovation also affected the participation configuration of this Conference. We created and generated daily social-demographic reports about the Youth Conference to understand the patterns of participation, and when some state or region was not participating as expected, and consequently lowering their chance to be heard by the government, we directly connected with local digital influencers to improve the promotion of the process in their region. These social-demographic reports and the gamified process later became a feature that every participatory process in Brasil Participativo could use.

In the National Culture Plan, we added a new, simplified layer of participation to gather the opinions of citizens who are not sufficiently involved in the process, which tends to be extensive and detailed. We opted to connect with a Brazilian open-source software called *Empurrando Juntas*[2], a tool where we can create affirmations, and we gathered users' perceptions of these affirmations (whether they agreed, disagreed, or were indifferent). From the user perception, the tool uses artificial intelligence to group the opinion using different personas, and cluster this opinion for the administration to understand what the vision of these different groups is about, an affirmation or theme.

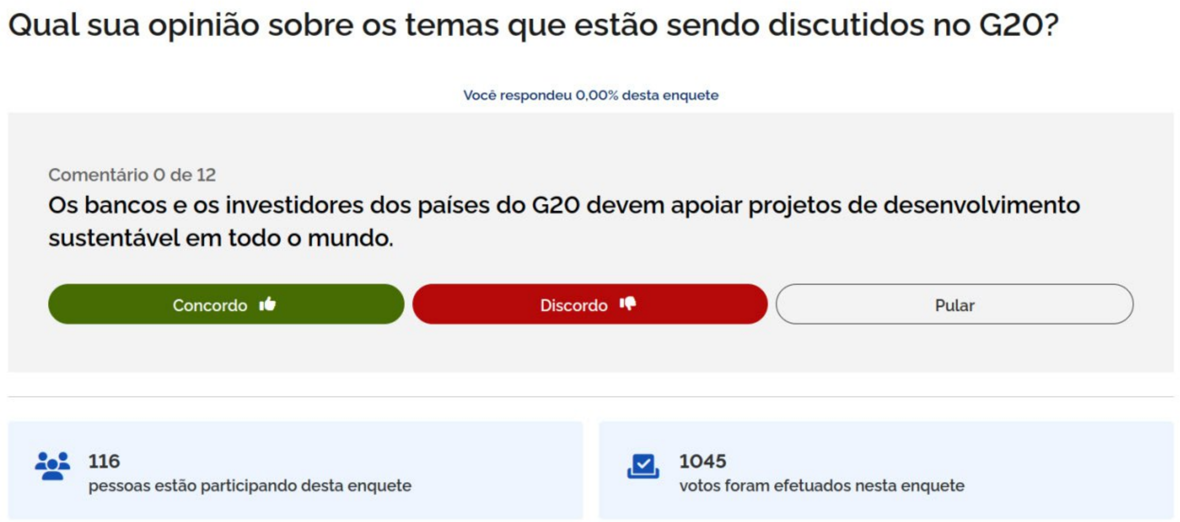
In Figure 4.2, we can see this tool used in the G20 Participativo process, which follows the same principle as the National Culture Plan. This rapid participation was positioned at the top of the process page as the user's first interaction, prompting them to share their opinion and helping them understand the process further. This created what we called the Participation Funnel, and later, we encouraged other Ministry portfolios to use the same technique.

Throughout the project and development of the platform, the LabLivre team was present in various onsite participatory processes (as shown in Figure 4.3), to understand how these processes occur, to collect the participants' perception of the stages of participation, and also about the Brasil Participativo platform, and used this data to improve even futher the technical and methodological aspect of the brazillian participation process.

Also, the partnership of the Instituto Cidade Democratica, the research group Resocie[3] of the Institute of Political Science of the University of Brasilia (IPOL-UnB), with financial support from Agence Française de Développement (AFD)[4], conducted different workshops with historically excluded groups for them to give feedback about the platform and the

[2]https://sobre.ejparticipe.org/ (Accessed on: 9 of April 2026)
[3]https://resocie.org/ (Accessed on: 9 of April 2026)
[4]https://www.afd.fr/fr (Accessed on: 9 of April 2026)

Figure (4.2) Opinion Survey applied in G20 Participativo Process to fast participation engagement



participatory process, so Brasil Participativo and public administrators could improve and allow better representation of these groups in the public policy-making.

This data enabled the Academia team to identify opportunities to use Artificial Intelligence to help not only citizens but also public administrators who use Brasil Participativo to make their processes available. In addition to a model to identify hateful speech in proposals or comments on the platform, we created a feature to recommend proposals that citizens may want to participate in, encouraging them to participate more. Also, we are developing an assistant to write proposals that could lead to more engagement in the voting process, and are likely to be accepted by the public administrator.

To improve the analysis of the participation data, we used Artificial intelligence to generate qualitative reports on the process, which the public administrator used as additional information when evaluating participation and further developing the public policy that emerged from the participatory process. Also, to the citizen, the tool will display how their contributions were incorporated into the final document of the public policy related to the participatory process.

All this innovation and contribution to improving public policy-making were developed in collaboration with external partners by the Lab-Livre, which had approximately 90 researchers involved. Most of the members are undergraduate students who had their first contact with a real product and stakeholder in the Brasil Participativo project. As part of the laboratory's mission, the senior researchers guided these undergraduate students in learning how to collect requirements, engage with users to understand their challenges, and think through and construct a technical solution, while maintaining the vision of improving the social participation of Brazilian citizens.

Figure (4.3) Directory and LabLivre members present in the 4th National Youth Conference

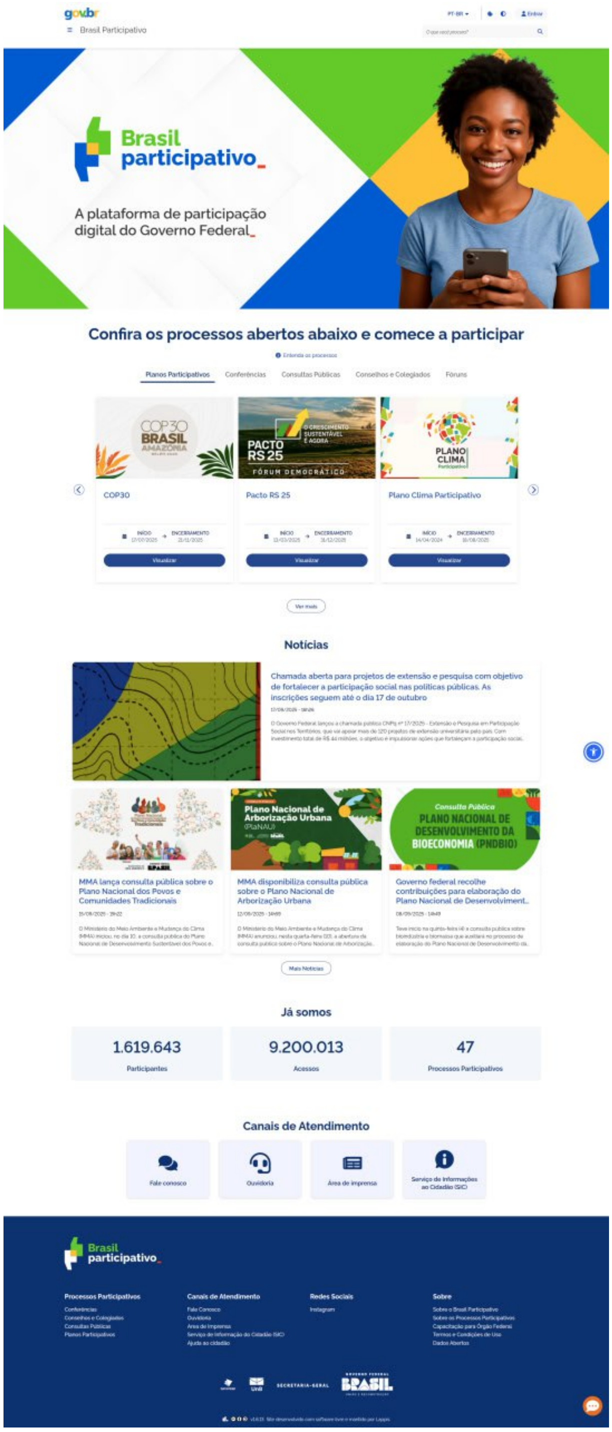
The Brasil Participativo was also present at various international events and settings. Beyond our presence at the Decidim annual event, Decidim Fest, we connected with this open-source community by integrating their Coordination Committee [5] to understand the community's needs regarding the platform, helping prioritize and develop solutions for the framework.

In 2024, the Brasil Participativo was selected by the OECD Observatory of Public Sector Innovation (OPSI) for the OECD Government-to-Government (Gov2Gov) Innovation Incubator. Partnering with New York City's Civic Engagement Commission (CEC) and the Greater London Authority, the Brasil Participativo team launched a challenge inviting people to explore how to promote broader and more equitable digital participation[6].

Brasil Participativo was also awarded second place in the largest public Information and Communication Technologies (ICT) in Brazil as the best Open Government solution of 2025 [7]. Also in 2025, it received the Regional and Thematic Award (Federal Strategy for Social Participation and Popular Education) of the first-ever Open Gov Challenge Awards promoted by the Open Government Partnership (OGP). [8]

[5] https://meta.decidim.org/processes/news/f/1719/posts/363 (Accessed on: 9 of April 2026)
[6] https://oecd-opsi.org/work-areas/gov2gov-innovation-incubator/ (Accessed on: 9 of April 2026)
[7] https://abep-tic.org.br/2025/08/08/gov-digital-2025-premia-solucoes-que-ja-mudam-a-vida-do-cidadao/ (Accessed on: 9 of April 2026)
[8] https://www.opengovpartnership.org/the-open-gov-challenge/open-gov-challenge-awards/#toc_2 (Accessed on: 9 of April 2026)

Figure (4.4) Actual Home Page of Brasil Participativo platform (as of March 2026)



Chapter 5

Results

The study of the Brasil Participativo project enabled the mapping of three distinct phases, analyzed through the lens of project and product management traceability. Since its launch, the project has operated under a multi-sectoral arrangement spanning two facets. In the **governance facet**, the UnB Laboratory (LabLivre) and the Social Participation Directorate of the General Secretariat of the Presidency of the Republic were responsible for strategic decisions. In the **technological facet**, software development was carried out by the laboratory using the open-source platform *Decidim* as its foundation, with additional support from Dataprev, which assumed responsibility for infrastructure and hosting, as well as database cross-referencing for process analysis and Business Intelligence (BI) construction (Silva Rocha Aguiar et al., 2024). While the laboratory focused on developing innovative features, partner organizations and companies contributed to the development of regular features and shared emerging technologies and practices with the lab.

This project began in May 2023 and continues as of this research (March 2026), with programming ending in November 2026. For the purpose of the research, it is considered a 34-month project (May 2023 to March 2026). During this period, the platform has **1.712.832 active unique users**, more than **9 million accesses** through the **205 participatory processes**[1].

The phases were defined for a unique project event or strategic participatory launch that changed the platform product's vision. **Phase 1** lasted 6 months, from May 2023 to December 2023. It was marked per the platform and the PPA process launch. With the success of the process and the platform's adoption, **Phase 2** marks a shift in perspective from project to product. It occurred from January 2024 to February 2025 (13 months) and began with the change in the Government team director, as well as processes such as the G20, the National Plan of Culture, and the National Climate Change Conference.

[1] Data from https://brasilparticipativo.presidencia.gov.br/ (Accessed on: 16 of March 2026)

Finally, **Phase 3** starts in February 2025 to present (March 2026), totaling 15 months, and starts with a strategic meeting to align the project's goals between the Academia and Government teams.

In the main repository, the project has over **113 release tags**, more than **4000 commits** and **676 merge requests** [2], without counting the adjacent module repositories[3].

For this analysis, the teams and the responsibilities of those involved are defined as follows:

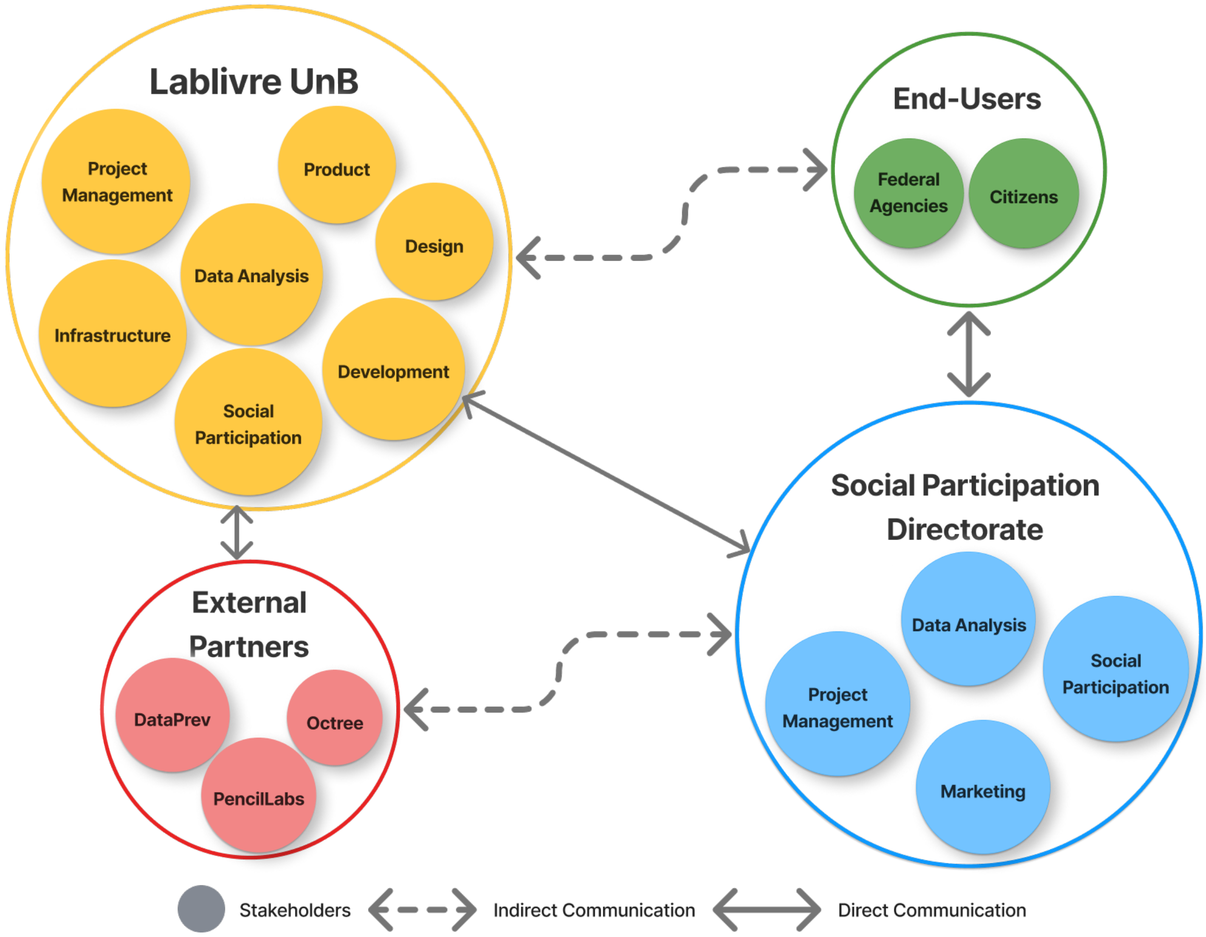
- **Social Participation Directorate**:
  - **Assignments**: responsible for helping collect, define requirements, and prioritize them in terms of the **political** aspect of the platform.
  - **Roles**: **Project Management**, **Marketing**, **Social Participation Specialist**, **Data Analysis**
- **Lablivre UnB**:
  - **Assignments**: are responsible for helping to collect, define requirements, and prioritize them in terms of the **technical** aspect of the platform. In addition, it helps develop and innovate in the social participation methodology.
  - **Roles**: **Project Management**, **Product**, **Social Participation Specialist**, **Design**, **Development**, **Infrastructure**, **Data Analysis**

Figure 5.1 shows the teams and project stakeholders. The teams on the Academia side changed over time, as shown in the figure. In this research, we treat the internal teams on the **Government** side as a single team. We are interested in the **Academia** teams and roles, their ability to collaborate with the government team, and practices they can use to improve this collaboration. The Academia teams actively contributed to requirements gathering and to building the participation methodology with the Government team, but most communication with the End-users was mediated by the Government team. The End-Users of Brasil Participativo are: Federal Agencies that make their participatory processes available, which we call **administrative users**; and the citizens who participate in these processes, the **common users**.

[2] Data from the main GitLab repository https://gitlab.com/lappis-unb/decidimbr/decidim-govbr (Accessed on: 16 of March 2026)

[3] https://gitlab.com/lappis-unb/decidimbr/components-brasil-participativo (Accessed on: 16 of March 2026)

Figure (5.1) Roles and Communication across Stakeholders



The project used various sources to track its progress. Figma files were used to document workflows and design components, and GitLab repositories were used for code version control and development tasks. Notion pages for internal documentation and task boards. We collected case material from sources and analyzed teams' knowledge of and use of that material.

In total, 42 case materials (called **Artifacts**) were analyzed and described in **Appendix-D** (Section 7.4) to substantiate the characterization of the phases and the identification of practices. They were categorized into 10 types, such as *meeting*, *diagram*, *presentation*, *document*, *spreadsheet*, *note*, *issue*, *commit*, and *board*. Figure 5.5 presents the proportion of evidence by type, and Table 5.1 shows an example of how the evidence categorization is presented in Table 7.4. When citing the **artifacts**, we use the **Evidence ID** (e.g. #T003DM Integrated Workflow - Phase 1). This evidence accounts for the teams' practices and partially answers **research question RQ2**.

Table (5.1) Example of the Evidence Categorization - Full table can be found in Appendix-D (Section 7.4)

Evidence ID	Evidency Name	Evidence Type	Present in Phase
#T003DM	Integrated Workflow - Phase 1	diagram	DevOps Phase 1, Product Phase 1
#T004DM	Roles and Responsibility Map - Phase 1	diagram	DevOps Phase 1, Product Phase 1
#T005DM	Retrospective Dev Team - 06/24	diagram	DevOps Phase 2, Product Phase 2

5.1 Data Analysis

For each phase of the project, evidence of DevOps practices (Díaz et al., 2024) and the **27 Product Management (SPM) problems** (Springer and Miler, 2022) was sought (Table 2.1). DevOps practices were mapped using the *codebook* from the article by (Díaz et al., 2024) (Table 7.1) and adapted to the context, adding an *ID*, yielding **21 codes** analyzed. For instance, codes related to infrastructure and automation were not considered, as the platform's infrastructure is maintained by the external partner, Dataprev (Silva Rocha Aguiar et al., 2024).

We classified the presence of the codes as explained in the Figure 3.2: ***Yes*** if the code exists; ***No*** if not identified; ***Partially*** if it exists but is not predominant; or *Not Applicable* (not relevant to the context). We did the same for the SPM problems categorized according to Springer and Miler (2022). To confirm the code or the existence of the problem, we related the researchers' evidence and perceptions to this code/theme.

An example of coding can be found in Figure 5.2, and the tables detailing all SPM practice codes and DevOps codes evaluation, with evidence, are in **Appendix-B and Appendix-C** (Section 7.2 and 7.5).

Figure (5.2) Example of DevOps practices content analysis for Brasil Participativo in Phase 1

Devops - Coding - Stage 1			
Taxonomy	Existence	Evidency	Note
blame	Yes	#T002MG	Review at each iteration.
communication/ collaboration	Yes	#T001MG #T002MG #T003MG	Communication and collaboration remained effective throughout the process, as all team members actively participated in meetings.
continuous improvement	Yes	-	Continuous improvement remains the team's primary objective.
cultural silos/conflicts	Partially	#T001IE #T001MG #T002MG #T003MG	While there is a somewhat siloed culture regarding management tool usage, decisions about meetings and demands involve the entire team.

5.2 Focus-Group Validation

After the first analysis and coding of the Brasil Participativo Product Management Problems and DevOps practices, we invited participants to validate the research's perception of that information. It was 8 focus group sessions, one per project phase and theme (Product Management problems and DevOps practices), with participants from the LabLivre-UnB team. The quantity of people per session varies, but at least one senior member of each core team (Product and Development) and one undergraduate intern from that team. We invite the seniors who played an active role in each phase of the project.

The Figure 3.3 presented in Subsection 3.1.3 shows the steps taken in the focus group validation. In the session, the researcher presented one code or problem at a time, commented on the evaluation, the **artifacts** supporting the practices, and the perceptions that guided it. Then, all participants were allowed to comment on the evaluations and artifacts, sharing their perceptions (whether they agreed or not). In the end, the researcher asked the group to reach a consensus on the occurrence. Excerpts from the session transcripts will be used to corroborate the analysis of the best practices.

Figure 5.3 and Figure 5.4 show the differences between the researcher's evaluation and the focus group's consensus on the codes. One can say that the researcher's perception was optimistic regarding the number of DevOps practices present (or partially present) throughout the project, as well as the relatively small number of problems. Information on the codes that changed their status can be found in the complementary material in Appendix-B and Appendix-C (Section 7.2 and 7.5).

Figure (5.3) Comparison of the quantity of Product Management occurrences. Especially in phase 3, the focus group strongly identified the presence of more problems.

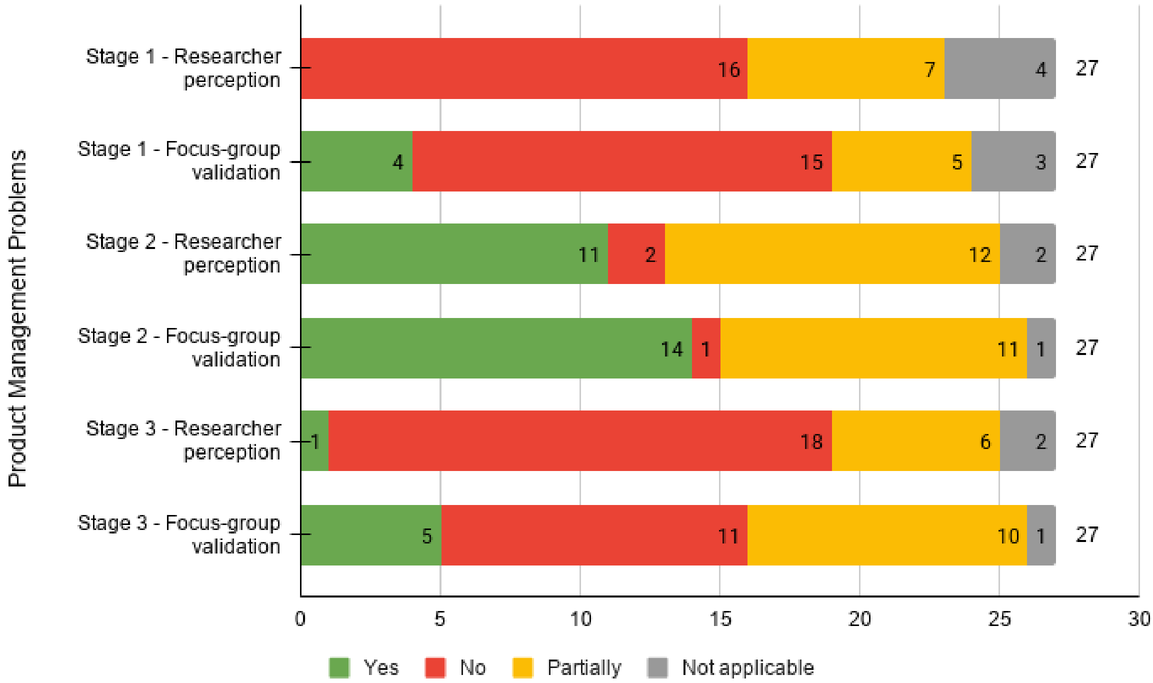
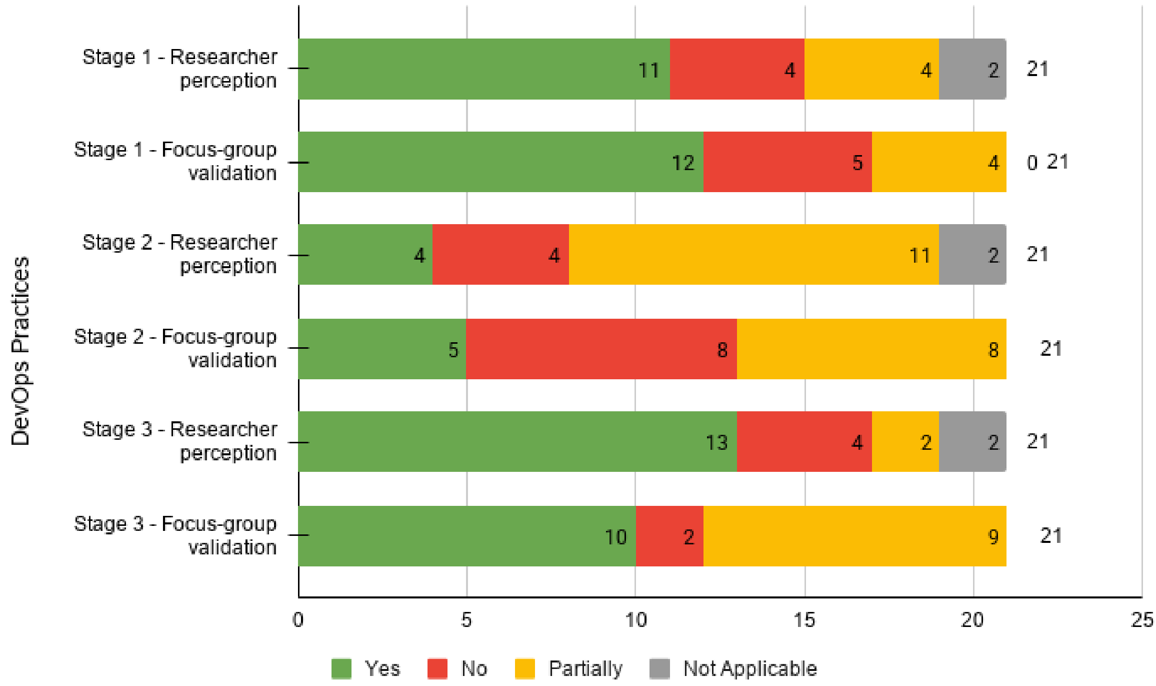


Figure (5.4) Comparison of the quantity of DevOps practices occurrences. In phase 2, the researcher obtained an optimistic perception, identifying more practices when the focus group classified them as not present.



5.3 Phase 1: PPA and Early Experiences (March to December 2023)

The first phase, from March to December 2023, was landmarked by the platform's launch in conjunction with the Participatory PPA (Plurianual Plan) process. The central objective of this participatory process was to validate a proof of concept for the participation platform, which would allow citizens to submit their proposals to the federal government. This process became one of the largest on the platform, prompting numerous proposals to be submitted directly to the federal government. Given the platform's public acceptance and initial success, it expanded to host two other major processes: the Youth and Food Security conferences.

At this phase, the team was small, and the platform's roadmap and priorities were still unclear. The focus was strictly on discovering the objectives and capabilities of the *Decidim* product, directly serving only three stakeholders: the Social Participation Directorate (partner) and the two agencies responsible for the conferences (administrative users).

The delivery needed to be quick due to political expectations and the PPA process deadline, so for the development, the Decidim tool **Snippet** was used. This tool allowed rapid changes to HTML, CSS, and JavaScript at specific points in the platform without affecting other pages or Decidim components. The drawback of this approach, however, is that each affected page is rendered twice: first by loading the original frontend, then by applying the snippet override, which introduces performance overhead and degrades the user experience. The Decidim framework was originally designed for European cities and small-scale groups, and adapting it for use at the federal level introduced significant performance limitations.

Requirements were discussed in meetings with all team members, including the government team. The focus was on the **citizen End-user**, and the Social Participation Specialist was responsible for collecting users' needs about participatory processes during conferences and plenary sessions, and then translating them into features for the Brasil Participativo platform.

The description of the government team is in Table 5.2, and the composition of the academic side (excluding the four professors responsible for the project coordination), including the number of senior and undergraduate students in each team, is in Table 5.3. All members from academia and government communicated through the Planning, Review, and Weekly Project Alignment meetings, during which they discussed requirements and priorities to ensure alignment with the project vision.

Table (5.4) Phase 1 - Summary of content analysis results.

Phase 1	Yes	No	Partially	Not Applicable
DevOps practices	12	5	4	0
Product Management Problems	4	15	5	3

In the Table 5.4, we can see the quantities of code identified in Phase 1, and their levels. Of the 21 DevOps practices analyzed, we found a strong presence of 12, most of them in the ***Culture***, ***Skills and Roles***, and ***Sharing*** categories, such as ***C01. blame***, ***C02. communication***, ***C03. collaboration***, and ***C04. continuous improvement***. This strong presence was largely due to evidence of weekly meetings involving all teams, such as Planning, Review, and Weekly Project Alignment meetings(#T001MG, #T002MG, and #T003MG). Also, ***S02. responsibility/ownership sharing***, ***S03. skills/knowledge sharing*** and ***SR02. role definition/ attributions*** codes have artifacts that define the responsibilities of team members (#T004DM) and how the team workflow flows from the arrival of a new requirement from the Government side to its implementation (#T002DM, #T003DM).

Table (5.2) Government team composition throughout Phases - It did not change in the duration of the project

Team	Responsibility
Director	Serves as the strategic liaison across government departments on participation matters and develops strategies based on new social participation methodologies.
Participation Specialist	The main stakeholder for digital participation; presents the platform to public agencies, gathers requirements grounded in social participation concepts and agencies' practical experiences, and is responsible for the transition from the previous platform.
Participation Facilitators	Deliver training and provide support to agencies in managing their participation processes.
Participa+ Developer	Developer of the previous platform.
Presidency's Quality Assurance	Stakeholders who support the validation of BP functionalities.
Participation Digital Communicator	Oversees the directorate's communications and also supports strategic process management.
Data Analyst	A stakeholder and data analyst focused on participation-related data; handles requests related to producing data and insights to support participation processes.

Most of the DevOps codes classified as **No** or **Partially** are from the ***Management*** and ***Organization Structure***. As an example, the ***M01. change management*** was classified as partially because although the code changes were versioned, some changes to the **snippet** code were not. As said in the focus group: *"Basically, the first six months of was implemented code using the snippet. Sometimes we didn't develop tests or document the updated version of the snippet."*

We did not identify the ***M03. product management***, since the team's focus was on the success of the first participatory process and on understanding the Decidim software.

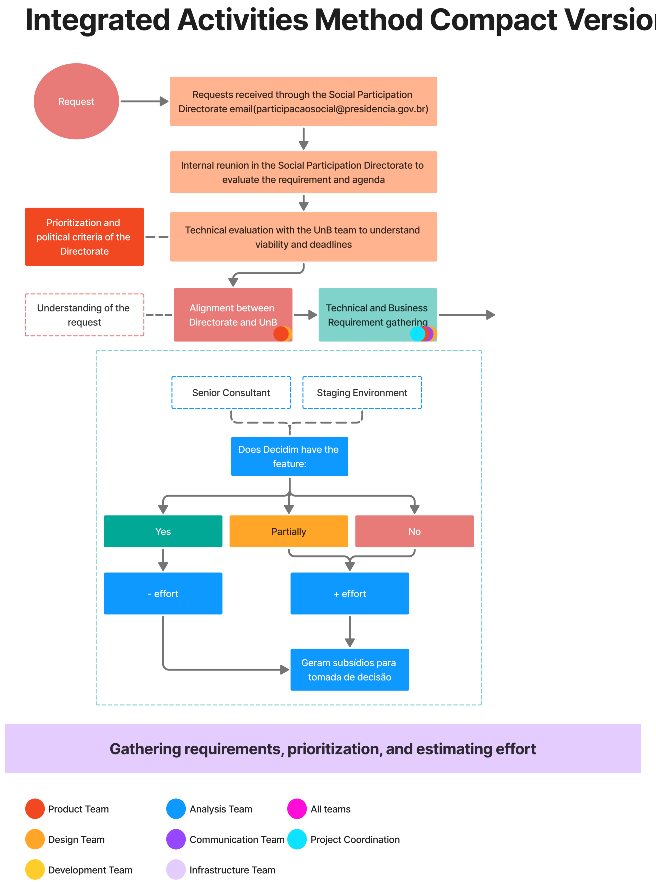
Table (5.3) Academia team composition in Phase 1

Teams	**Senior**	**Undergraduate Students**
Development	2	5
Participation Specialist	4	1
Design	3	1
Data Analysts	0	3
Infrastructure	1	1
Total	**10**	**11**

There was no specific product team or defined **Product Manager** role. The PM code related to the product management team (***P43. Lack of trust in the product team (micro-management)***) was also identified as Not applicable.

Figure 5.5 presents an excerpt from the original Integrated Activities Method diagram, highlighting which team acts in the state where the demand is located.

Figure (5.5) Excerpt from the Integrated Activities Method diagram - Phase 1



Regarding *Product Management* problems, this phase has the lowest problem-occurrence rate. Out of the 27 listed in the literature Springer and Miler (2022), 5 are present partially (Table 5.4), and 4 are strongly identified.

P12. Low software quality, ***P6. Lack of continuous integration and delivery (impact on 'time to market')***, ***P41. High expectations from external partners, which are not possible to be met***, ***P85. Lack of automated testing***, and ***P9.Technical debt*** is traced to the rapid need to generate value and to the use of the **snippet** to solve code problems without deployment. When deployment is necessary, the code changes are directly integrated into *Decidim* rather than following the proposed Module approach. This led to technical debt at every phase of the project.

Also, ***P51. Roadmap focused on features instead of goals and business value*** was classified as partially, due to the general knowledge of the goal with the development of the platform, although it did not have a formal roadmap. ***P43. Lack of trust in the product team (micro-management)*** was **Not applicable** because it did not have a product team at the time.

It is important to mention that the PM code ***P64. Working in silos***, and DevOps code ***C05. cultural silos/conflicts*** were not identified in this phase due to the intense communication across all teams, as mentioned in the meeting artifacts. An important context is that the team was small, and as said *"We work in a joint effort to solve a problem."*

The condensation of practices, roles, and artifacts from this phase is found in Table 5.5.

Table (5.5) Phase 1 - Teams and Artifacts

Teams	**Artifacts**	
Development	**Meetings**	Planning (#T002MG)
Design		Review - all team members (#T002MG)
Participation Specialist		Weekly Project Alignment - all team members (#T001MG, #T003MG)
Infrastructure	**Diagrams**	Integrated Workflow (#T002DM, #T003DM)
Data Analysis		Roles and Responsibility Map (#T004DM)

5.4 Phase 2 - January 2024 to February 2025

The second phase, lasting 1 year and 2 months, began with the confirmation of the platform's success and a change in management at the Social Participation Directorate. The new strategic understanding was that Brasil Participativo could become the central platform for all social participation models. The new objectives were to replace the existing "Participa+ Brasil" platform, convince federal agencies to adopt it for their processes, and increase user traffic. To achieve this, teams adapted existing functionality to support the features in "Participa+".

This period was marked by structural tension. One major challenge was designing, developing, and deploying new functionalities while using them in an ongoing participatory process. This caused pressure on the development workflow, with rapid development cycles and a short time to test and validate these features. Consequently, it is a period of intense instability on the platform and a continuous need for critical bug fixes.

While three major **strategic participatory processes** were used as study objects to generalize and customize functionalities, the decision to provide personalized service to these **strategic participatory processes** conflicted with the product vision of an autonomous platform for administrative clients.

Table (5.6) Academia team composition in Phase 2

Teams	**Senior**	**Undergraduate Students**
Development	3	15
Product Manager (Participation Specialists)	2	0
Product Owner	2	5
Design	3	2
Data Analysts	2	6
Machine Learning	2	1
Infrastructure	1	7
Total	**15**	**36**

The architecture of Decidim (Figure 2.4) and the use of Decidim components among Brasil Participativo (Figure 2.3) show that the components/modules (features) are interchangeable, so one change in this module affects all processes that use it. We needed to understand how the participatory processes worked on the previous platform to replicate their functionality in the Decidim structure, but this was hampered by the different requirements of the strategic participatory processes. This clash of requirements led to various product management and development challenges.

The team's size is growing rapidly to handle the number of new improvements to be implemented on the platform, passing from 10 senior members and 11 undergraduate students to 15 and 36, respectively, as shown in Table 5.6. A product team was created with **Product Manager** (PM) and **Product Owner** (PO) roles. However, the PM was also the participation specialist. This dual role led to a conflict: the PM raised demands for customization of **strategic participatory processes**, while the PO had to write requirements based on client/product priorities, often clashing with the need for generalization.

Table (5.7) Phase 2 - Summary of content analysis results.

Phase 2	Yes	No	Partially	Not Applicable
DevOps practices	5	8	8	0
Product Management Problems	14	1	11	1

This phase had the lowest DevOps code strongly identified. As presented in Table 5.7, there were 5 codes as Yes, 8 codes not identified, and 8 partially identified. ***C04. continuous improvement***, ***M04. team self-organization & autonomy*** and ***S04. stack & tools sharing*** were strongly identified in all 3 phases, so the other 2 codes identified were ***C05. cultural silos/conflicts*** and ***OS03. organizational silos/conflicts***.

Those codes carry a negative connotation. In phase 1 ***C05. cultural silos/conflicts*** were not present, the team was small, and all team members participated in meetings, understanding the project goals. ***OS03. organizational silos/conflicts*** were partially because, although it has constant communication, the physical silo exists because of different workplaces.

But, in Phase 2, ***C05. cultural silos/conflicts*** were strongly identified due to the lack of an integrated meeting to align the requirements and goals. Meetings registered in the artifacts were limited to a single team (the product team), and the Weekly Project Alignment meeting lacked the product owner, hindering requirements gathering. It lacked a workflow and was directed to the development team without prioritization. Also, the Data Analysis and Infrastructure team is reported as a siloed team, following their own priorities, which delayed some deliveries (*"The Data/ML team are in their own bubble, that don't communicate with others. They have their own cultures and practices. There's no knowledge exchange with others, and in the end, it harms the project."*). In the focus group session, it was also mentioned that the Design team was a silo: *"We didn't have contact with the Data and Design team. We only sent the requirement, or sometimes, we didn't send it at all."*

For the ***OS03. organizational silos/conflicts***, it was mentioned that the tentative plan was to separate work teams into squads, with members from design, product, and development. This tentative failed due to the lack of a senior member in all squads; also, the squad was sometimes responsible for a strategic process rather than a Decidim component. This resulted in requirements that overlap with other squad requirements due to the reuse of components throughout different processes.

The focus group decided that ***C02. collaboration***, ***C03. communication*** was not present in this phase. As mentioned, we didn't meet with the product and development team, and the requirements don't include a defined flow. As commented in focus group: *"For example, the requirements were not being properly communicated; there was no clear communication of the requirements, right? They were passed on to third parties, who then distributed these requirements without proper alignment to understand them and relay them to the development team. Even the handoff of these requirements was unclear. Once structured, they were sent directly to the person responsible for the distribution network. So, I would say that [the code] almost didn't exist.""*

Concerning the PM codes, it was also the phase with the most PM problems strongly and partially identified (**14 yes** and **11 partially**). Among them are:

- **P35:** Strategy and priorities are changing frequently.
- **P43:** Lack of trust in the product team (micro-management).
- **P64:** Working in silos (problem with communication, synchronization between teams).
- **P74:** Determining the true value of the product that the customer needs

During the focus group session, participants repeatedly noted that the priorities of the requirements changed (***P35***), that it was difficult to obtain a clear definition of those requirements, and that they aligned with all requirements from the different stakeholders. Remembering the context of this phase, the academia teams were trying to improve the platform for strategic participatory processes and also generalize it to handle the functionalities of the previous platform. Also, the academia team had difficulty identifying the development priority; they tried to implement all features at once without considering the best solution. The lack of alignment meetings to define priorities was detailed in the ***P74*** code.

P43: Lack of trust in the product team (micro-management) existed during a period of maturation of the product team structure. The requirement was passed directly to the Development lead rather than to the Product Owner. One focus group participant who belonged to the product team at the time reported that the perception of trust in their work was due to the fact that they had belonged to the development team before: *"One thing that gave them [the presidency team] the confidence to delegate the task was that I had been part of the development team. [...] And gradually, they began to understand that I was actually part of the product team."*

Also present in the DevOps code, the ***P64: Working in silos*** was strongly commented on during the focus group session. Again, the lack of alignment, planning meetings, clear definitions of responsibilities, and a product vision contributed to this evaluation. The feature delivery was accompanied by communication confusion, and the workflow was not defined.

"When we picked up tasks to work on and encountered an error or had some doubt about how something should function—whether related to workflow or functionality—we would already feel discouraged, because we knew it would take one or two days to get in touch with the person who had documented it."

"The workflow pipeline, for me, was not really well defined."

Another 11 problems appeared partially, such as (***P82. Product Manager role not clearly defined***), where the role was understood merely as requirements gathering and status reporting, lacking authority over the demand flow. Due to this diagnosis at the end of phase 2, a new Integrated Workflow and Roles and Responsibilities Map was created using a strategic participatory process as a case.

The artifact analysis showed only **meeting artifacts** of one team (#T005MG, #T006MG, #T007MG). Roadmaps were created by each squad when this structure was in place (#T007DM, #T008DM, #T001DT), without the collaborative alignment. The Integrated Workflow and Roles and Responsibilities map found was incomplete and not implemented in the phase (#T006DM, #T009DM, #T010DM).

A significant difference between Phases 1 and 2 was the lack of a clear definition of work methodology and roles.

The definitions of roles and artifacts for Phases 2 are found in Table 5.8 and Table 5.9.

Table (5.8) Comparison of Roles across Phases 1 and 2

Roles	Phase 1	Phase 2
Participation Specialist	✓	✓
Product Manager		✓ (focus in strategics processes)
Product Owner		✓ (without reponsibilities definied)
Designer	✓	✓
Development	✓	✓
Infrastructure	✓	✓
Project Management	✓	✓
Data Analyst	✓	✓

Table (5.9) Comparison of Artifacts across Phases 1 and 2

Artifacts	Phase 1	Phase 2
Weekly Project Alignment	All members (#T001MG, #T003MG)	Without a Product Owner (#T007MG)
Planning	All members (#T002MG)	Per team/squad (#T005MG, #T006MG)
Review	All members (#T002MG)	Per team/squad (#T005MG)
Retrospective	-	Only one register (#T005DM)
Integrated Workflow	All teams (#T002DM, #T003DM)	Not applied in the phase / Incomplete (#T006DM, #T009DM)
Roles and Responsibility Map	All teams (#T004DM)	Not applied in the phase / Incomplete (#T010DM)
Roadmap	-	Per team/squad (#T007DM, #T008DM, #T001DT)
Project Planning	-	#T001ST
Task Board	-	Per team/squad (#T005BD, #T006BD, #T007BD)

5.5 Phase 3 (February 2025 – March 2026)

The third phase, currently lasting 15 months, is marked by an alignment meeting between the laboratory and the presidency. Following feedback from administrative users that

Table (5.10) Academia team composition in Phase 3

Teams	Senior	Undergraduate Students
Development	5	9
Product Manager (Participation Specialists)	2	4
Product Owner	1	3
Design	2	4
Data Analysts	3	6
Machine Learning	5	1
Infrastructure	1	7
Technical Writing / Internal Communication	2	2
Total	**22**	**36**

some customized functionalities were not performing adequately, the directorate and the laboratory management decided to rethink the product offering. The objective shifted to maturing the platform code and restructuring functionalities that had been customized only for specific processes. The product vision is now consolidated, and improvements are less experimental and focused on established functionalities.

Strategic planning documentation (#T002BD), integrated Workflow (#T013DM), and roles and responsibilities (#T014DM) were re-created. A **strategic participatory process** was used to pilot and validate this new workflow. The team composition maintained similar to the previous phase (see Table 5.10), and the roles were clarified:

- **Product Manager (Participation Specialist):** Acts in **strategic participatory processes** not to perform customizations, but to ensure processes are carried out correctly and to improve administrative user adoption.
- **Product Owner:** Assists in identifying generic requirements and assumes responsibility for understanding client priorities, translating them into tasks for the development and design teams.

A unified board was created containing all team tasks, organized by delivery priority. Urgent solution demands are mapped to a **Bug Board** (#T008BD), while improvement suggestions not in the roadmap go to a **Suggestions Board** (#T003BD). Delivery planning meetings now involve all teams (#T009MG), and the product team leader participates in weekly client alignment meetings (#T008MG) to gather information and translate it for the rest of the squad.

This effort is evident in the DevOps code: the only two missing items are DevOps-related code, and, since we did not have a dedicated DevOps team, they were evaluated as absent across all phases of the project. The results of the code in this phase are presented

Table (5.11) Phase 3 - Summary of content analysis results.

Phase 3	Yes	No	Partially	Not Applicable
DevOps practices	10	2	9	0
Product Management Problems	5	11	10	1

in Table 5.11, which show a total of 10 strongly identified codes and 9 partially identified codes.

Code such as ***C02. collaboration***, ***C03. communication***, ***C06. culture, values & best practices***, ***C05. cultural silos/conflicts***, ***M05. transfer of work between teams***, and ***OS03. organizational silos/ conflicts*** improved, and reports generated by the application of integrated planning, meetings, workflows, and integrated task boards.

"[Before], at most, it was just the demand page on GitLab, and that was it. But it evolved so much that now we even have joint planning sessions between product and development."

"Now I think that with this integrated board, everything is in sync. Before, we would communicate with them directly on Telegram."

It is important to mention that in this phase, the code ***OS02. enabler (platform) team*** appears for the first time, with a different interpretation. Here, the enabler team was the Technical Writing team. This team is responsible for documenting the use of the platform by End-User Federal bodies, helping them create their processes on the platform by directly requesting the presidency. It removes one of the many responsibilities of the product and the government team, allowing them to focus on other tasks.

SPM problems now appear in a total of 5 strongly and 10 partially identified. Code related to the Product Manager role, such as ***P82. Product Manager role not clearly defined and communicated in the organization*** and ***P8. Different expectations about product management communication per stakeholder (high/low level)*** were solved, since it now has the Roles and Responsibility Map (#T014DM).

P51. Roadmap focused on features instead of goals and business value was solved, and ***P35. Strategy and priorities are changing frequently,*** improved, although it still occurs due to the political nature of the project.

Some problems arise from user data, and it is noted that, throughout the project, a better data analytics setup and direct communication between Academia and End-users (citizens and Federal bodies) could improve the product overall.

To condense the first analysis of this research, Table 5.12 presents the comparison of **roles** in each project phase, and the Table 5.13 brings comparison lists of the **artifacts** generated throughout the project lifecycle.

Table (5.12) Comparison of Roles across Phase 1, 2, and 3

Roles	Phase 1	Phase 2	Phase 3
Participation Specialist	✓	✓	✓
Product Manager		✓ (focus in strategics processes)	✓
Product Owner		✓ (without reponsibilities definied)	✓
Designer	✓	✓	✓
Development	✓	✓	✓
Infrastructure	✓	✓	✓
Project Management	✓	✓	✓
Data Analysis	✓	✓	✓
Technical Writer		✓	✓

5.6 Discussion

The first research question is **RQ1. How do different continuous delivery practices and team configurations affect the product management of a software in a government-academia collaboration context?**

As observed in Chapter 5, throughout the project phases, different team configurations and practices were adopted, leading to varying levels of implementation and distinct problem identification. Table 7.2 and Table 7.3 summarize all identified problems and DevOps practices, as shown in Figure 5.6, and the Table 5.12 describes the team configuration of each phase, answering this question.

Throughout the project, several issues were addressed, such as ***P1. No company strategy***, ***P51. Roadmap focused on features instead of goals and business value***, and ***P82. Product Manager role not clearly defined and communicated in the organization*** (Table 5.14). Establishing a joint roadmap and clearly defining an issue requiring age, as well as for the Product Owner, helps maintain the product vision and sustain priorities in day-to-day work.

Besides, having the product team responsible for requirements elicitation reduces friction between development teams and stakeholders, improves context understanding, and prevents the development team from being overloaded with non-development tasks. In the context of Brasil Participativo, this role becomes even more important, as it is responsible for understanding how a demand may impact different processes, given that features often span multiple workflows due to shared components.

Some practices remained present throughout the project (see Table 5.15); however, ***C01. blame***, ***M05. transfer of work between teams***, and ***S02. responsibility/ownership sharing*** varied in intensity over time. The lack of clarity in task assignments, along with a limited understanding of the Decidim structure, led to these practices

Table (5.13) Comparison of Artifacts across Phases 1, 2, and 3

Artifacts	Phase 1	Phase 2	Phase 3
Weekly Project Alignment	All members (#T001MG, #T003MG)	Without a Product Owner (#T007MG)	With a Product Owner (#T008MG)
Planning	All members (#T002MG)	Per team/squad (#T005MG, #T006MG)	All academia teams (#T009MG)
Review	All members (#T002MG)	Per team/squad (#T005MG)	All academia teams (#T009MG)
Retrospective	-	Only one register (#T005DM)	Monthly - All academia teams (#T011DM)
Integrated Workflow	All teams (#T002DM, #T003DM)	Not applied in the phase / Incomplete (#T006DM, #T009DM)	All academia teams (#T002ST, T013DM)
Roles and Responsibility Map	All teams (#T004DM)	Not applied in the phase / Incomplete (#T010DM)	All teams (#T014DM)
Roadmap	-	Per team/squad (#T007DM, #T008DM, #T001DT)	All academia teams (#T012DM)
Project Planning	-	#T001ST	#T002BD
Task Board	-	Per team/squad (#T005BD, #T006BD, #T007BD)	All academia teams (#T001BD)
Requests and Bug Report Boards	-	-	All teams (#T002ST, #T008BD)
Product/Releases announcements	-	-	#T001CN

Table (5.14) Improved Product Management Problems at the end of the Project

Product Management Problems	Phase 1	Phase 2	Phase 3
P1. No company strategy	Partially	Partially	No
P51. Roadmap focused on features instead of goals and business value	Partially	Yes	No
P82. Product Manager role not clearly defined and communicated in the organization	Yes	Partially	No

being less effectively applied during Phase 2. A lack of trust in transferring responsibility was also reported as a recurring issue.

The codes highlighted in Table 5.16 require further in-depth analysis, given their results. Although all of them showed some level of improvement, it was only partial. Codes such as ***C01. blame***, ***C02. collaboration***, ***C03. communication***, ***M05. transfer of work between teams***, and ***S02. responsibility/ownership sharing*** either ceased to exist or worsened due to the lack of a clearly defined workflow and well-established roles and responsibilities during Phase 2.

Despite efforts such as alignment meetings and integrated task boards, communication with the government team remained an ongoing issue.

Reports such as: "[Before], I think the communication mostly happened between the leads. And then it really became quite segregated." and

"[Before], at most, it was just the demand page on GitLab, and that was it. But

Table (5.15) DevOps Practices present in all Phases

DevOps Practices	Phase 1	Phase 2	Phase 3
C01. blame	Yes	Partially	Partially
C04. continuous improvement	Yes	Yes	Yes
M04. team self-organization & autonomy	Partially	Yes	Yes
M05. transfer of work between teams	Yes	Partially	Partially
S01. alignment of dev & ops goals	Yes	Partially	Yes
S02. responsibility/ownership sharing	Yes	Partially	Partially
S03. skills/knowledge sharing	Yes	Partially	Yes
S04. stack & tools sharing	Yes	Yes	Yes
SR01. cross-functionality/skills	Yes	Partially	Yes
SR02. role definition/attributions	Yes	Partially	Yes

Table (5.16) DevOps Practices that have worsened over time

DevOps Practices	Phase 1	Phase 2	Phase 3
C01. blame	Yes	Partially	Partially
C02. collaboration	Yes	No	Partially
C03. communication	Yes	No	Partially
C05. cultural silos/conflicts *	No	Yes	Partially
M05. transfer of work between teams	Yes	Partially	Partially
OS03. organizational silos/conflicts *	No	Yes	Partially
S02. responsibility/ownership sharing	Yes	Partially	Partially

it evolved so much that now we even have joint planning sessions between product and development." Reinforce the importance of alignment meetings between teams. These should not be limited to Product, Design, and Development teams; they should also involve adjacent teams, such as Data and Infrastructure, which are still reported to operate in silos.

The existence of a joint planning process and roadmap with the government partner is essential to avoid recurring shifts in priorities. Due to the product's governmental nature, it is inherently subject to changes driven by political factors. However, when the partner understands the product vision and the software development workflow, defining which requirements to prioritize becomes easier. The creation of a dedicated board to handle non-urgent demands also helped the Academia team better understand and manage priorities.

The codes ***OS03. Organizational silos/conflicts***, ***C05. Cultural silos/conflicts***, and ***P64. Working in silos (problems with communication and synchronization between teams)*** deserves special attention, as their final result remained only partially improved due to the challenges related to differing ways of working across institutions.

"For example, designers work in one way, right? Infrastructure works in another way, and even the way they communicate differs. So, in my view, it does exist. We try to reduce these communication gaps, but I think they persist. Reinforcing that, I think within the lab it still happens, but certainly even more so with the presidency."

The lack of a clear definition of responsibilities within the Academia team can lead to a loss of trust from the partner. In Phase 2, when internal teams struggled to communicate and transfer work effectively, issues such as ***P43. Lack of trust in the product team (micro-management)*** and ***P13. The team's lack of motivation due to a lack of understanding of why they are doing things—resulting in low commitment to achieving goals—also emerged***. This was largely driven by frequent rework caused by misalignment of requirements:

"When thinking about other factors contributing to a lack of motivation, I would point much more to rework, because it was something very recurrent. People would do a huge amount of work, and sometimes it was very difficult—technically difficult—and in the end, due to a lack of alignment, they had to redo it."

It is evident that inter-institutional challenges persist. The university promotes students to help develop its projects under the guidance of senior members. Here, the priority is not fast delivery to stay ahead of the market, but to reach a point where students are autonomous enough to meet expectations and to encourage new challenges that improve their learning. The intention of arranging a collaborative development in which both parties are partners and build the project's scope together is to keep challenges and innovation alive, fostering continuous learning.

There is a misleading perception in the government's organizational culture that co-development with the university is primarily about service provision rather than collaborative development. So, when evaluating the problems that emerged, we have to keep in mind that the partnership's goal is to promote learning and innovation, develop well-built software, identify problems together, and create new perspectives for a digital service. It is important to gather and prioritize requirements, but also to respect the political fluctuations that may arise.

The vision of academia not delivering is also mentioned in Wen et al. (2020) and Siqueira et al. (2018), where adopting an autonomous continuous delivery pipeline was one of the proposals to overcome the government's bias toward low productivity in collaborative projects with academia.

We suggest transparency-promoting artifacts (project alignment meetings, release channels, collaboratively built roadmaps) to help reduce friction between government and academia, break down silos, and build trust in the Academia team's work. Having a clear definition of roles and responsibilities, not only on the academia side but also on the government side, is also important.

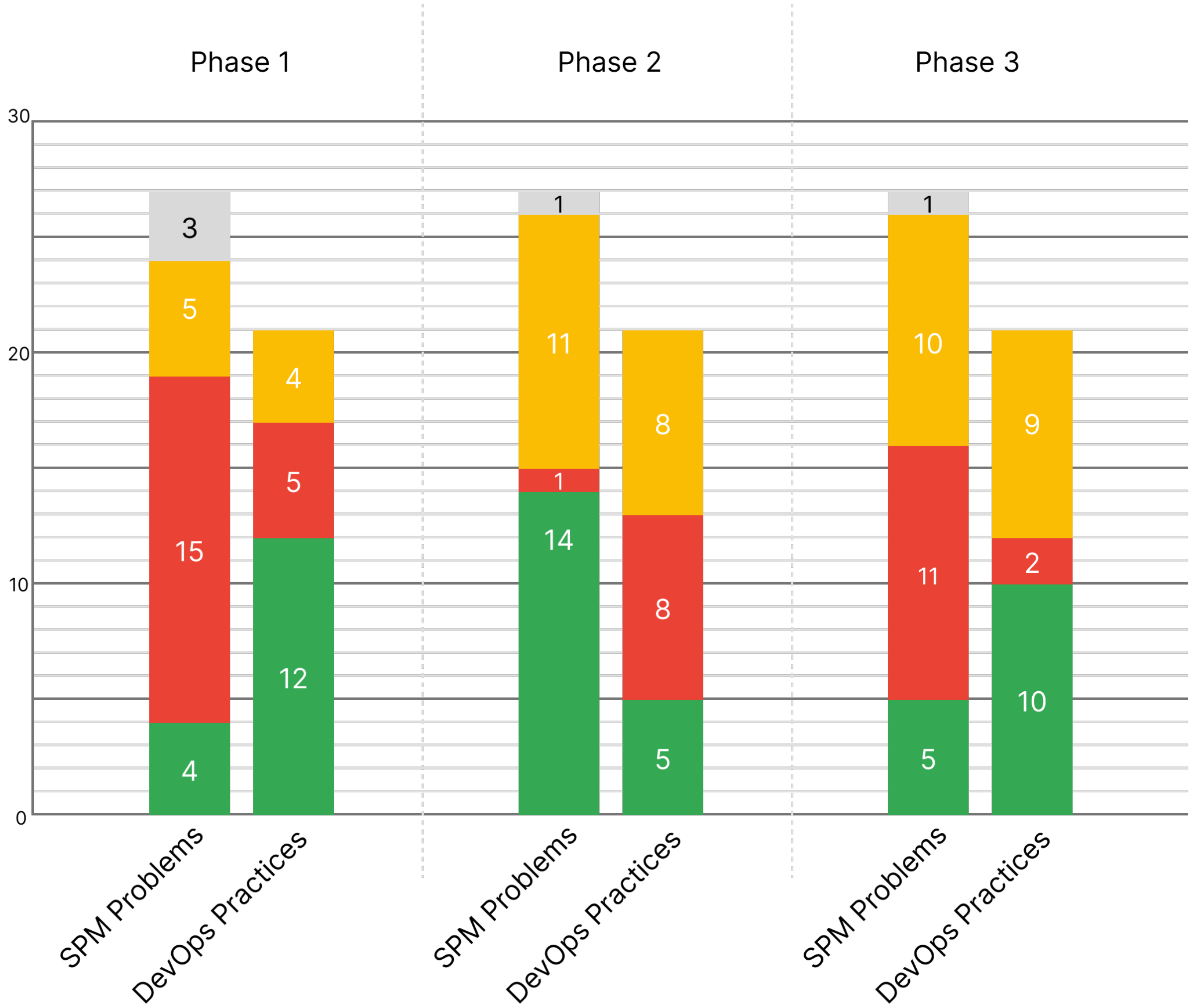
"Partnership, from their perspective, is seen as a service provision of our work. In the view of some of them, it is purely about executing what they want, the way they want it."

The Table 5.17 shows problems that persisted in all phases of the project. A key issue throughout the project was the lack of continuous integration and automated testing (as reflected in ***P6. Lack of continuous integration and delivery—impacting time to market, P85. Lack of automated testing***, and ***M01. Change management***). This is largely due to the structure of the infrastructure services contracting arrangement. Under this contract, access to production and staging environments is restricted and must always be mediated by Dataprev, which directly impacts the ability to build and maintain deployment pipelines.

"Because, in fact, to me the issue with Dataprev is that it is a poorly designed contract, right?"

The hyphothesis ***"Low adoption of continuous delivery best practices can lead to more product management problems, and specific artifacts and team structures can help ensure the applicability of these practices"*** can be confirmed, since that is in phase 2 (Figure 5.6), where we identify most of the product management problems, and the least DevOps practices are implemented. Also, this is the phase where it lacks the use of artifacts for communication (e.g., planning and reviews)

Figure (5.6) The quantity of DevOps practices and SPM problems identified throughout the project, divided into phases. Phase 2 shows the most identified problems and the lack of DevOps practices.



and for sharing responsibility (e.g., an integrated task board, an integrated workflow, and role and responsibility maps). This phase lacks a defined Product manager/owner role responsible for guiding the teams in maintaining the product vision as they define priorities.

As a response to **RQ1**, the Figure 5.7, 5.8, and 5.9 show how the DevOps practices mapped as present, partially present, and absent by the study affected the team structures and their interaction during the project. These mappings are used as a basis for the theory constructors and propositions of the Díaz et al. (2024) article.

Table (5.17) Product Management Problems that persisted throughout the Project

Product Management Problems	Phase 1	Phase 2	Phase 3
P6. Lack of continuous integration and delivery (impact on 'time to market')	Partially	Partially	Partially
P85. Lack of automated testing	Yes	Partially	Partially
P41. High expectations from external partners, which are not possible to be met	Yes	Yes	Partially
P9. Technical debt	Yes	Partially	Yes

Figure (5.7) Relationship between DevOps Practices and Team Structures of Phase 1 - adapted from Díaz et al. (2024) and with indicatives of presence based on the study analysis

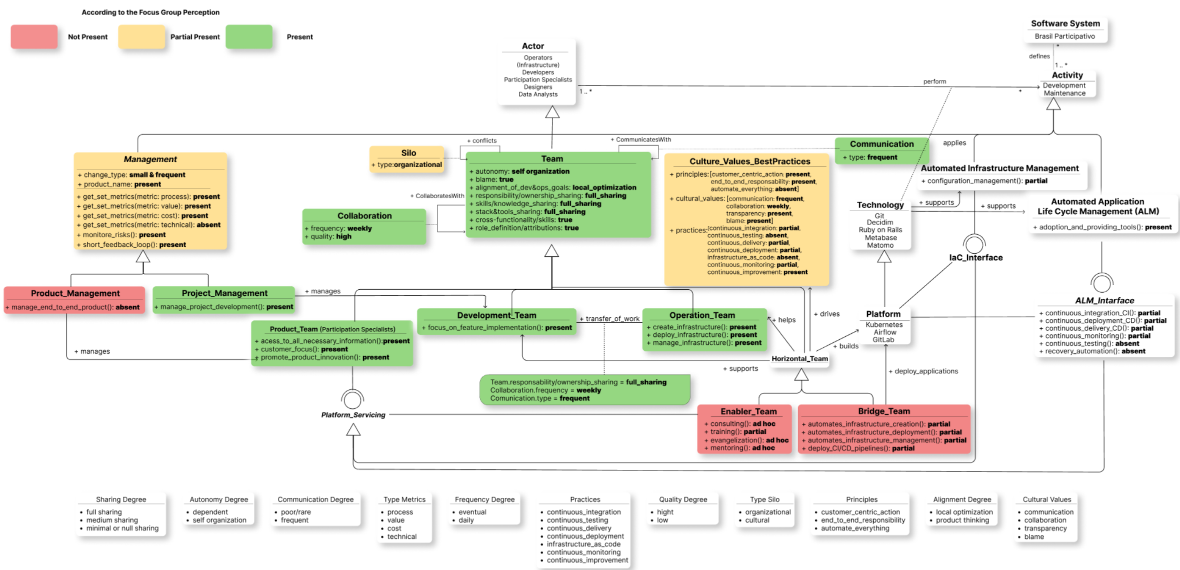


Figure (5.8) Relationship between DevOps Practices and Team Structures of Phase 2 - adapted from Díaz et al. (2024) and with indicatives of presence based on the study analysis

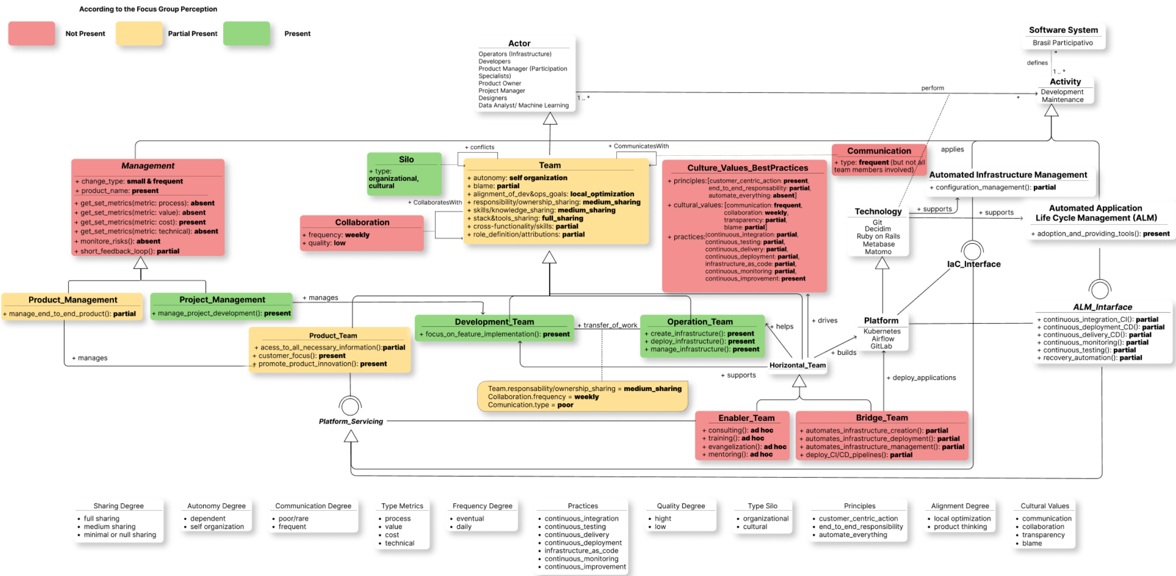
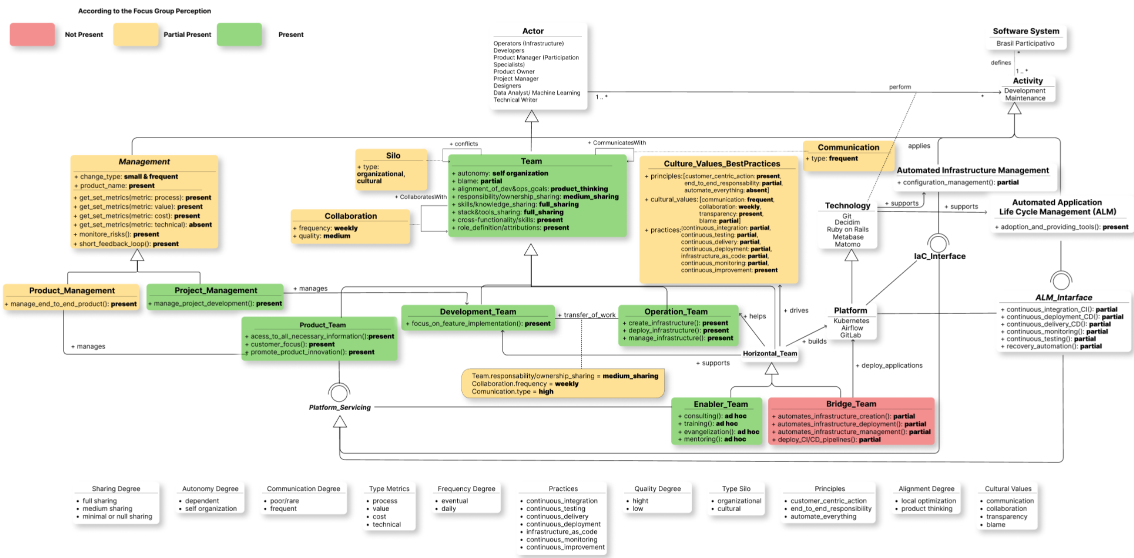


Figure (5.9) Relationship between DevOps Practices and Team Structures of Phase 3 - adapted from Díaz et al. (2024) and with indicatives of presence based on the study analysis



In response to **RQ2. What practices and team structures favor the development of a government-academia collaborative product?** We identified that a well-defined product structure, combined with continuous alignment practices, can foster effective collaboration. A team structure should include a **Product Manager**, responsible for the long-term product vision, and a **Product Owner**, focused on requirements elicitation and day-to-day interaction with stakeholders. These two roles can ensure continuous engagement with the government partner and with internal academia teams, including development, design, data, and infrastructure. It is important to note that this role should not act as a bottleneck but as a facilitator of continuous delivery, ensuring alignment with product goals and promoting shared solution-building across teams.

Practices such as **planning**, **review sessions**, and **regular project alignment meetings** with the government partner are necessary to maintain communication, collaboration, and expectation management. These can help mitigate misalignment and reduce rework in contexts where requirements are dynamic due to political factors. A collaboratively constructed **roadmap** enables both to better understand priorities and adapt to changes while maintaining a unified vision of value delivery.

Another aspect relevant to improving co-development is the explicit definition of roles, responsibilities, and workflows. The use of **role and responsibility maps** and clear process definitions (**Integrated Workflow**) helps reduce organizational and cultural silos, facilitates knowledge transfer, and strengthens ownership across teams. In this sense, transparency-promoting artifacts, such as **integrated task, request, and bug report boards**, also **channel/media to announce software releases**, can help build trust and enable smoother interactions between academia and government.

The presence of an **enabler team**, such as a technical writer team, keeps knowledge dissemination to both the government partner and end users. This helps reduce dependence on specific individuals and enhances the product's sustainability beyond the project lifecycle.

Following these valuable contributions, the final team configuration and artifacts that can help foster collaboration between government and academia are shown in Table 5.18 and Table 5.19. Each role and artifact offers specific benefits that help maintain transparency, improve communication, clarify responsibilities, and keep government and academic teams aligned throughout the project.

Table (5.18) Recommended Team Structure to improve Product Management in Government-Academia Software Co-development

Recommended Team Structure	Benefits
Product Manager	Responsible for maintaining the long-term product vision and ensuring strategic alignment between the product goals and the objectives of the government-academia partnership.
Product Owner	Focused on requirements elicitation and day-to-day interaction with stakeholders. This role helps ensure continuous engagement with the government partner and alignment with internal academia teams.
Designer	Responsible for UI/UX design, creating visual representations of workflows and interfaces to improve shared understanding and user experience.
Development	Implements technical solutions and product features, ensuring that prioritized requirements are effectively translated into working software.
Infrastructure	Establishes the technical environment necessary for deployment and monitoring, ensuring the product's reliability and operational continuity.
Project Management	Monitors project progress and manages the academia-government relationship through communication adapted to the institutional context.
Data Analysis	Supports decision-making by providing evidence-based insights into product impact and usage through systematic data analysis.
Technical Writer	Supports knowledge dissemination to both the government partner and end users. This reduces dependence on specific individuals and enhances the product's sustainability beyond the project lifecycle.

Table (5.19) Recommended Artifacts and Practices to improve Product Management in Government-Academia Software Co-development

Recommended Artifact or Practice	Benefits
Weekly Project Alignment	Supports communication between the government and academia partners, with the participation of the Product Owner and Product Manager, helping both sides understand development priorities and the overall product direction.
Planning	Aligns internal academia teams, such as development, design, and data analysis, with government priorities and supports backlog restructuring according to current needs.
Review	Enables stakeholders to validate implemented features, ensuring that technical deliveries meet expectations and provide the intended value.
Retrospective	Supports continuous improvement by identifying what worked well, what needs to be improved, and how future deliveries can be better organized.
Integrated Workflow	Describes the demand flow from intake to production delivery. It maps the actors involved in each phase, how academia and government interact, and the expected outputs of each stage.
Roles and Responsibility Map	Makes transparent to the government where each academic role acts and what its responsibilities are. It also helps internal teams understand how each role contributes to delivering value.
Roadmap	Provides a visual timeline of features and priorities, aligning expectations between partners regarding the product's evolution and delivery plan.
Project Planning	Offers a high-level strategic view of the partnership, defining the key milestones and deliverables to be achieved during the project lifecycle.
Integrated Task Board	Helps internal teams maintain transparency and visibility over ongoing tasks, while also supporting product control and monitoring for the government partner.
Request Board	Provides a centralized space where the government partner can register demands, which can then be structured, prioritized, and allocated into the roadmap.
Bug Report Board	Provides a centralized space where the government partner can report bugs. These issues may require urgent resolution and can override previously prioritized weekly tasks when necessary.
Product and Release Announcements Channel	Maintains the government partner engagement by providing regular updates on product releases, ensuring transparency throughout the software lifecycle.

Chapter 6

Conclusion

This study investigated how team structures and DevOps practices influence product management in a government–academia collaborative context, using the Brasil Participativo project as an exploratory case study. Regarding **RQ1. How do different continuous delivery practices and team configurations affect product management in a government–academia collaboration context?** The results show that team structures and practices directly impact product management effectiveness.

Phases that included practices that fostered integrated teams, defined workflows, and continuous alignment sessions (such as Phase 1 and especially Phase 3) demonstrated greater adoption of DevOps practices and lower Product Management problems, resulting in better outcomes in terms of communication, reduced silos, and improved prioritization of requirements, confirming the hypothesis shared before. In contrast, Phase 2 was marked by a lack of alignment rituals and unclear definitions of responsibility, and saw a substantial increase in product management problems, including frequent priority shifts, a lack of trust in the product team, and difficulties defining product value.

Also, it is observed that collaborative culture, shared responsibility, and structured communication mechanisms directly influence how product management challenges emerge and are addressed.

Regarding **RQ2. What practices and team structures favor the development of a government–academia collaborative product?**, The study identifies that successful collaboration is supported by a combination of well-defined roles, integrated workflows, and transparency-promoting artifacts.

The presence of a Product Manager focused on strategic alignment and a Product Owner responsible for operationalizing requirements could be essential to bridge the gap between stakeholders and teams. Additionally, practices such as joint planning sessions, review meetings, and weekly alignment with the government partner are intended to maintain a shared understanding and reduce rework.

Software development is a collaborative effort; tasks may be distributed across different teams, and these tasks must be managed and ordered according to specific criteria (Alsaqqa et al., 2020), so the artifacts, such as integrated task boards, roadmaps, role and responsibility maps, and workflow diagrams, can help improve coordination, reduce silos, and increase trust between institutions.

In this study, we observe a behavior previously described by Wen et al. (2018, 2020) that government–academia collaborations introduce specific challenges, including shifts in priorities driven by political factors, cultural differences between institutions, and constraints on infrastructure and deployment processes.

Addressing these challenges requires not only technical practices but also organizational mechanisms and both partners' willingness to adopt a product and project management vision to create a collaborative software that endures beyond the project deadline.

6.1 Future work

Our focus in this research was on identifying the occurrence of Product Management Problems, as described by Springer and Miler (2022). Further investigation into solutions to those problems can be a future research topic in government-academia settings. It could serve as a foundation for the strategies proposed in Springer et al. (2023). Mapping the SPM problems identified in this case study to the solution space described in the literature would enable evaluation of which strategies are most effective in government–academia contexts. Also, it could identify gaps where existing approaches may be insufficient.

Another promising direction is investigating user-centered, data-driven product management practices in government platforms, particularly how user analytics and feedback loops can better inform decision-making. Since it was a flaw in the project, collecting user feedback and using this data to guide development are among the benefits of the Continuous delivery method. A study focusing on user satisfaction with observing product management decisions and development performance can provide insights into what citizens may need in e-government software.

The study of social participation through digital platforms extends beyond software engineering and can contribute to the technical-political literature.

6.2 Thread to Validity

The threats to validity in this research stem from the study's qualitative design and the researcher's involvement in the case context.

Given the author's active participation in the Brasil Participativo project as a product team member, this can bring researcher bias. This dual position may have influenced data interpretation and the selection of evidence. To address this risk, focus group sessions with multiple project participants were incorporated to consider diverse perspectives and challenge the researcher's initial interpretations. Although it only considered the Academia perspective of the project, if it had included the government side, it could have led to different perspectives. Also, while the triangulation with the focus group enhanced the credibility of the findings, it did not fully eliminate potential bias.

Another limitation is the shared context of the focus group participants. As all participants were involved in the project, their perceptions may have been shaped by personal experiences. This could potentially lead to overly critical or highly context-dependent interpretations of events. This dynamic may have led to an emphasis on internal challenges that are not generalizable to other contexts.

There are also present threats to data completeness, as portions of the analysis relied on retrospective data collection. Some artifacts and perceptions were reconstructed after 34 months of the project. This may introduce recall bias and information loss. As said in the research design, some historical logs were not fully available for all phases, and it was necessary to reconstruct from archived materials and memory. This limitation could have affected the accuracy of the evidence used in the analysis.

This research relies on a single case study, which could limit the external validity and generalizability of the findings. The Brasil Participativo project has unique characteristics, including its governmental context, technopolitical orientation, and multi-institutional collaboration, that may not be directly transferable to other domains.

Methodological subjectivity is inherent in the use of qualitative coding and directed content analysis. Although the established codebooks (DevOps practices (Díaz et al., 2024) and SPM problems (Springer and Miler, 2022)) and validation through focus groups were employed, the interpretation of evidence and classification of codes may still reflect subjective judgment.

Chapter 7

Appendices

All data related to this research is in Appendix A. DevOps Practice Codebook (Section 7.1) reference by Díaz et al. (2024), Appendix-B. SPM final evaluation (Section 7.2), Appendix-C. DevOps Practices final evaluation (Section 7.5) and Appendix- D. Evidence (Section 7.4) is available at `https://bit.ly/dissertation_data_bruna_pinos` as supplementary material.

7.1 Appendix - A. DevOps practices Codebook

Table (7.1) The DevOps Practice codebook used in this research was adapted from Díaz et al. (2024). Original codebook available in the supplementary material.

Domain	Code
CULTURE	C01. blame
	C02. collaboration
	C03. communication
	C04. continuous improvement
	C05. cultural silos/conflicts
	C06. culture, values & best practices
MANAGEMENT	M01. change management
	M02. metrics, visibility & feedback
	M03. product management
	M04. team self-organization & autonomy
	M05. transfer of work between teams
ORGANIZATIONAL STRUCTURE	OS01. devops (bridge) team
	OS02. enabler (platform) team
	OS03. organizational silos/conflicts
SHARING	S01. alignment of dev & ops goals
	S02. responsibility/ownership sharing
	S03. skills/knowledge sharing
	S04. stack & tools sharing
SKILLS & ROLES	SR01. cross-functionality/skills
	SR02. role definition/attributions
	SR03. training, evangelization and mentoring

7.2 Appendix - B. Product Management Problems - Final Evaluation

Table (7.2) Final Evaluation Product Management Problems - Full table with Researcher's note, and Focus group quotations are available in the supplementary material.

Product Management Problems - Final Evaluation	Phase 1	Phase 2	Phase 3
P78. Lack of skills to use and analyze the data	**No**	**No**	**No**
P18. Lack of market research, no understanding of the business and trends in the industry	No	Partially	No
P19. Teams are not Agile, they just follow rules and do not use experimentation and a learning process	No	Partially	No
P27. Unqualified team members (individuals)	No	Partially	No
P69. Balancing between reactive and proactive work. When comparing hypotheses with facts, hypothesis loses in value to facts (such as clients' requests, bugs). Managing requirements instead of identifying problems and opportunities, seeking innovation.	No	Partially	No
P44. Teams are not autonomous and self-organized (difficult to organize, dispersed both responsibility and decision-making)	No	Yes	No
P8. Different expectations about product management communication per stakeholder (high/low level)	No	Yes	No
P43. Lack of trust in the product team (micro-management)	Not Applicable	Yes	No
P1. No company strategy	Partially	Partially	No
P51. Roadmap focused on features instead of goals and business value	Partially	Yes	No
P82. Product Manager role not clearly defined and communicated in the organization (what the role is about, what the responsibilities and objectives are, decisiveness)	Yes	Partially	No
P56. Price management is always experimentation burdened with risk	Not Applicable	Not Applicable	Not Applicable
P10. Lack of user analytics data	No	Partially	Partially
P13. The team's lack of motivation due to a lack of understanding of why they are doing things – no commitment to achieve goals	No	Yes	Partially
P35. Strategy and priorities are changing frequently	No	Yes	Partially
P64. Working in silos (problem with communication, synchronization between teams)	No	Yes	Partially
P74. Determining the true value of the product that the customer needs	No	Yes	Partially
P12. Low software quality	Partially	Partially	Partially
P26. Lack of support for research (no resources allocated to the team)	Partially	Yes	Partially
P6. Lack of continuous integration and delivery (impact on 'time to market')	Partially	Partially	Partially
P85. Lack of automated testing	Yes	Partially	Partially
P41. High expectations from external partners, which are not possible to be met	Yes	Yes	Partially
P25. Limited access to users in order to do research	No	Yes	Yes
P3. Wrong data analytics setup	No	Yes	Yes
P48. Lack of user research	No	Yes	Yes
P16. Lack of synchronization between product management and marketing and sales units	Not Applicable	Yes	Yes
P9. Technical debt	Yes	Partially	Yes

7.3 Appendix - C. DevOps Practices - Final Evaluation

Table (7.3) Final Evaluation of DevOps Practices - Full table with Researcher's note, and Focus group quotations are available in the supplementary material.

DevOps Practices - Final Evaluation	**Phase 1**	**Phase 2**	**Phase 3**
C01. blame	Yes	Partially	Partially
C02. collaboration	Yes	No	Partially
C03. communication	Yes	No	Partially
C04. continuous improvement	Yes	Yes	Yes
C05. cultural silos/conflicts	No	Yes	Partially
C06. culture, values & best practices	Partially	No	Partially
M01. change management	Partially	No	Yes
M02. metrics, visibility & feedback	Partially	No	Partially
M03. product management	No	Partially	Yes
M04. team self-organization & autonomy	Yes	Yes	Yes
M05. transfer of work between teams	Yes	Partially	Partially
OS01. devops (bridge) team	No	No	No
OS02. enabler (platform) team	No	No	Yes
OS03. organizational silos/conflicts	Partially	Yes	Partially
S01. alignment of dev & ops goals	Yes	Partially	Yes
S02. responsibility/ownership sharing	Yes	Partially	Partially
S03. skills/knowledge sharing	Yes	Partially	Yes
S04. stack & tools sharing	Yes	Yes	Yes
SR01. cross-functionality/skills	Yes	Partially	Yes
SR02. role definition/attributions	Yes	Partially	Yes
SR03. training, evangelization and mentoring	No	No	No

7.4 Appendix-D. Case Study Evidences

Table (7.4) All Evidence Collected in this research - Full table with description and links are available in the supplementary material

Evidence ID	Evidency Name	Evidence Type	Present in Phase
#T001MG	Weekly follow-up meetings	meeting	DevOps Phase 1, Product Phase 1
#T002MG	[Review | Planning] All teams	meeting	DevOps Phase 1, Product Phase 1
#T003MG	[Checkpoint] All teams	meeting	DevOps Phase 1, Product Phase 1
#T001IE	#574 "Identification of the 60 delegates in CONJFUVE"	issue	DevOps Phase 1, Product Phase 1
#T001DM	Youth Conference Journey	diagram	DevOps Phase 1, Product Phase 1
#T002DM	Integrated Activities Method - Pocket - phase 1	diagram	DevOps Phase 1, Product Phase 1
#T003DM	Integrated Activities Method - Complete - phase 1	diagram	DevOps Phase 1, Product Phase 1
#T004DM	Roles and Responsibilities - phase 1	diagram	DevOps Phase 1, Product Phase 1
#T005DM	Development Retrospective 06/24	diagram	DevOps Phase 2, Product Phase 2
#T004MG	Daily - Laboratory leaders	meeting	DevOps Phase 2, Product Phase 2
#T005MG	Product Review/Planning	meeting	DevOps Phase 2, Product Phase 2
#T006MG	Product Planning	meeting	DevOps Phase 2, Product Phase 2
#T007MG	Presidency Alignment Meeting	meeting	DevOps Phase 2, Product Phase 2
#T002IE	#987 "Verify sending of invitation email to ADMIN agency users" at lappis-unb / DecidimBR / ecossistema	issue	DevOps Phase 2, Product Phase 2
#T001ST	Planning_2024 Part 2	spreadsheet	DevOps Phase 2, Product Phase 2
#T006DM	Squad Structure	diagram	DevOps Phase 2, Product Phase 2
#T007DM	Participatory text roadmap	diagram	DevOps Phase 2, Product Phase 2
#T008DM	Events squad roadmap	diagram	DevOps Phase 2, Product Phase 2
#T001DT	Events squad delivery list	document	DevOps Phase 2, Product Phase 2
#T009DM	Work method - Squads	diagram	DevOps Phase 2, Product Phase 2
#T010DM	Roles and Responsibilities - phase 2	diagram	DevOps Phase 2, Product Phase 2
#T011DM	Monthly Retrospective	diagram	DevOps Phase 3, Product Phase 3
#T008MG	Brasil Participativo Project Coordination Meeting (UnB-SG)	meeting	DevOps Phase 3, Product Phase 3
#T009MG	[BP Web] Integrated Planning	meeting	DevOps Phase 3, Product Phase 3
#T001BD	Integrated Task Board	board	DevOps Phase 3, Product Phase 3
#T002BD	Project management	board	DevOps Phase 3, Product Phase 3
#T012DM	Roadmap 2/2025	diagram	DevOps Phase 3, Product Phase 3
#T002ST	Bug Management	spreadsheet	DevOps Phase 3, Product Phase 3
#T013DM	Work method - phase 3	diagram	DevOps Phase 3, Product Phase 3
#T002DT	Brasil Participativo Product Documentation Page	document	DevOps Phase 3, Product Phase 3
#T003DT	Training team activity management	document	DevOps Phase 3, Product Phase 3
#T014DM	Responsibility map	diagram	DevOps Phase 3, Product Phase 3
#T003BD	Requests and suggestions for improvements	board	DevOps Phase 3, Product Phase 3
#T003IE	Project Management - Action: Perform convergence of legacy BP versions with Decidim software	issue	Product Phase 3, DevOps Phase 3
#T004BD	Board and explanation about the Mobile product	board	Product Phase 3, DevOps Phase 3
#T015DM	Roadmap 1/2025	diagram	Product Phase 3, DevOps Phase 3
#T005BD	Development Task Board - 2/2024	board	DevOps Phase 2, Product Phase 2
#T006BD	Design Task Board - Design improvements - 2/2024	board	DevOps Phase 2, Product Phase 2
#T007BD	Design Task Board - Platform - 2/2024	board	DevOps Phase 2, Product Phase 2
#T001CN	Product Update - Telegram Channel	communication	Product Phase 3, DevOps Phase 3
#T016DM	Work Method - Brasil sem Fome	diagram	DevOps Phase 2, Product Phase 2
#T008BD	Bug Board	board	Product Phase 3, DevOps Phase 3

7.5 Appendix - E. Brasil Participativo Team Characterization according to the Focus Group Perception

Full-size images are available in the supplementary material.

References

Agile Alliance (2001). Principles behind the agile manifesto. https://agilemanifesto.org/iso/en/principles.html. Accessed: 2026-01-. Cited on p. 8.

Ahmed, S. K., Mohammed, R. A., Nashwan, A. J., Ibrahim, R. H., Abdalla, A. Q., M. Ameen, B. M., and Khdhir, R. M. (2025). Using thematic analysis in qualitative research. *Journal of Medicine, Surgery, and Public Health*, 6:100198. Cited on p. 27.

Alsaqqa, S., Sawalha, S., and Abdel-Nabi, H. (2020). Agile software development: Methodologies and trends. *International Journal of Interactive Mobile Technologies (iJIM)*, 14(11):246–270. Cited on pp. 13 and 67.

Anthopoulos, L., Reddick, C. G., Giannakidou, I., and Mavridis, N. (2016). Why e-government projects fail? an analysis of the healthcare.gov website. *Government Information Quarterly*, 33(1):161–173. Cited on pp. 1 and 17.

Barandiaran, X. E., Calleja-López, A., Monterde, A., and Romero, C. (2024). *Decidim, a Technopolitical Network for Participatory Democracy: Philosophy, Practice and Autonomy of a Collective Platform in the Age of Digital Intelligence*. SpringerBriefs in Political Science. Springer Nature Switzerland, Cham. Cited on pp. x, 21, and 22.

Braun, V. and Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101. Cited on pp. 27 and 28.

Brereton, P., Kitchenham, B., Budgen, D., and Li, Z. (2008). Using a protocol template for case study planning. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, EASE'08, page 41–48, Swindon, GBR. BCS Learning & Development Ltd. Cited on p. 25.

Büyüközkan, G. and Arsenyan, J. (2012). Collaborative product development: a literature overview. *Production Planning & Control*, 23(1):47–66. Cited on pp. 3 and 17.

Chen, L. (2015). Continuous delivery: Huge benefits, but challenges too. *IEEE Softw.*, 32(2):50–54. Cited on pp. 2, 8, and 9.

Chow, T.-M. and Cao, D.-B. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, 81(6):961–971. Agile Product Line Engineering. Cited on p. 13.

da Silva, F. B., Jaccoud, L., and Beghin, N. (2005). Políticas sociais no brasil: participação social, conselhos e parcerias. In *Questão Social e Políticas Sociais no Brasil*

Contemporâneo, chapter 8. Instituto de Pesquisa Econômica Aplicada (Ipea), Brasília. Cited on p. 19.

Davis, A., Bersoff, E., and Comer, E. (1988). A strategy for comparing alternative software development life cycle models. *IEEE Transactions on Software Engineering*, 14(10):1453–1461. Cited on p. 13.

Díaz, J., López-Fernández, D., Pérez, J., and González-Prieto, Á. (2021). Why are many businesses instilling a devops culture into their organization? *Empirical Software Engineering*, 26(2):25. Cited on pp. 2, 13, and 14.

Díaz, J., Pérez, J., Alves, I., Kon, F., Leite, L., Meirelles, P., and Rocha, C. (2024). Harmonizing devops taxonomies — a grounded theory study. *Journal of Systems and Software*, 208:111908. Cited on pp. x, xi, xiii, 3, 4, 14, 16, 25, 28, 41, 60, 61, 62, 68, 69, and 70.

Ebert, C. (2014). Software product management. *IEEE Software*, 31(3):21–24. Cited on p. 8.

Ebert, C. and Brinkkemper, S. (2014). Software product management – an industry evaluation. *Journal of Systems and Software*, 95:10–18. Cited on pp. x, 2, 9, and 10.

Edmonds, W. A. and Kennedy, T. D. (2017). *An Applied Guide to Research Designs: Quantitative, Qualitative, and Mixed Methods*. SAGE Publications, Thousand Oaks, CA, 2nd edition. Cited on p. 27.

Eriksson, M. (2011). The history and evolution of product management. Accessed: 2025-08-31. Cited on p. 10.

Geracie, G. T. and Eppinger, S. D. (2013). *The Guide to the Product Management and Marketing Body of Knowledge (ProdBOK® Guide)*. AIPMM, Mountain View, CA, 1st edition. Cited on p. 10.

Goldfinch, S. (2007). Pessimism, computer failure, and information systems development in the public sector. *Public Administration Review*, 67(5):917–929. Cited on p. 17.

Governo Federal do Brasil (2025). Consultas públicas — participa + brasil. Cited on p. 19.

Guntamukkala, V., Wen, H. J., and Tarn, J. M. (2006). An empirical study of selecting software development life cycle models. *Human Systems Management*, 25(4):265–278. Cited on p. 13.

Hsieh, H.-F. and Shannon, S. E. (2005). Three approaches to qualitative content analysis. *Qualitative Health Research*, 15(9):1277–1288. PMID: 16204405. Cited on p. 28.

Humble, J. and Farley, D. (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley, Boston, MA. Cited on p. 8.

Kellner, D. (2001). Globalisation, technopolitics and revolution. *Theoria: A Journal of Social and Political Theory*, (98):14–34. Cited on p. 18.

Kittlaus, H.-B. and Fricker, S. A. (2017). *Software Product Management and Pricing: Key Success Factors for All Software Organizations*. Management for Professionals. Springer, Berlin, Heidelberg, 2nd edition. Cited on pp. 2, 9, 10, and 11.

Kon, F., Meirelles, P., Lago, N., Terceiro, A., Chavez, C., and Mendonca, M. (2011). Free and open source software development and research: Opportunities for software engineering. In *Proceedings of the 2011 25th Brazilian Symposium on Software Engineering*, SBES '11, page 82–91, USA. IEEE Computer Society. Cited on p. 17.

Kontio, J., Bragge, J., and Lehtola, L. (2008). The Focus Group Method as an Empirical Tool in Software Engineering. In *Guide to Advanced Empirical Software Engineering*, pages 93–116. Springer, London. Cited on pp. 4, 23, and 30.

Kontio, J., Lehtola, L., and Bragge, J. (2004). Using the focus group method in software engineering: obtaining practitioner and user experiences. In *Proceedings. 2004 International Symposium on Empirical Software Engineering, 2004. ISESE '04.*, pages 271–280. Cited on p. 30.

Kurban, H. C., Peña-López, I., and Haberer, M. (2017). What is technopolitics? a conceptual schema for understanding politics in the digital age. *IDP: Revista d'Internet, Dret i Política*, 24. Cited on pp. 18 and 19.

Leite, L., Kon, F., Pinto, G., and Meirelles, P. (2020). Platform teams: An organizational structure for continuous delivery. In *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, ICSEW'20, page 505–511, New York, NY, USA. Association for Computing Machinery. Cited on pp. 2 and 9.

Leite, L., Pinto, G., Kon, F., and Meirelles, P. (2021). The organization of software teams in the quest for continuous delivery: A grounded theory approach. *Information and Software Technology*, 139:106672. Cited on pp. 14 and 15.

Leite, L., Rocha, C., Kon, F., Milojicic, D., and Meirelles, P. (2019). A survey of devops concepts and challenges. *ACM Comput. Surv.*, 52(6). Cited on pp. 2, 13, and 14.

Luz, W. P., Pinto, G., and Bonifácio, R. (2018). Building a collaborative culture: a grounded theory of well succeeded devops adoption in practice. In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '18, New York, NY, USA. Association for Computing Machinery. Cited on pp. 3, 13, and 14.

Luz, W. P., Pinto, G., and Bonifácio, R. (2019). Adopting devops in the real world: A theory, a model, and a case study. *Journal of Systems and Software*, 157:110384. Cited on pp. 3, 13, and 14.

Lwakatare, L. E., Kuvaja, P., and Oivo, M. (2016). An exploratory study of devops: Extending the dimensions of devops with practices. In *ICSEA 2016: The Eleventh International Conference on Software Engineering Advances*, pages 91–97. IARIA. Cited on p. 13.

López-Fernández, D., Díaz, J., García, J., Pérez, J., and González-Prieto, (2022). Devops team structures: Characterization and implications. *IEEE Transactions on Software Engineering*, 48(10):3716–3736. Cited on p. 2.

Macarthy, R. W. and Bass, J. M. (2020). An empirical taxonomy of devops in practice. In *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 221–228. Cited on p. 9.

Maglyas, A., Nikula, U., and Smolander, K. (2013). What are the roles of software product managers? an empirical investigation. *Journal of Systems and Software*, 86(12):3071–3090. Cited on p. 11.

Meirelles, P., Wen, M., Terceiro, A., Siqueira, R., Kanashiro, L., and Neri, H. (2017). Brazilian public software portal: an integrated platform for collaborative development. In *Proceedings of the 13th International Symposium on Open Collaboration*, OpenSym '17, New York, NY, USA. Association for Computing Machinery. Cited on pp. 3 and 17.

Mergel, I. (2015). Open collaboration in the public sector: The case of social coding on github. *Government Information Quarterly*, 32(4):464–472. Cited on p. 17.

Ministério da Cultura (2025). Conheça o plano nacional de cultura. Acesso em: 23 jun. 2025. Cited on p. 19.

Runeson, P. and Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2):131–164. Cited on pp. 24 and 25.

Scholl, H. (2003). E-government: a special case of ICT-enabled business process change . In *2014 47th Hawaii International Conference on System Sciences*, volume 10, page 12 pp., Los Alamitos, CA, USA. IEEE Computer Society. Cited on pp. 1 and 17.

Schwaber, K. and Sutherland, J. (2020). The 2020 scrum guide. `https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf`. Scrum Guides, accessed 30 August 2025. Cited on pp. 11 and 13.

Seaman, C. (1999). Qualitative methods in empirical studies of software engineering. *IEEE Transactions on Software Engineering*, 25(4):557–572. Cited on p. 27.

Seaman, C. B. (2008). Qualitative methods. In Shull, F., Singer, J., and Sjøberg, D. I. K., editors, *Guide to Advanced Empirical Software Engineering*, pages 35–62. Springer London, London. Cited on pp. 26 and 27.

Secretaria-Geral da Presidência da República (2014). Cartilha da política nacional de participação social (pnps). Cited on p. 19.

Silva Rocha Aguiar, C., Alves, I., Gomes, L., Pinos, B., Bellix, L., and Parra, H. (2024). Colaboração multissetorial para desenvolvimento e manutenção de soluções tecnológicas de participação: o caso do brasil participativo. *Revista do Serviço Público*, 75(a):89 – 107. Cited on pp. x, 3, 6, 18, 25, 32, 33, 38, and 41.

Siqueira, R., Camarinha, D., Wen, M., Meirelles, P., and Kon, F. (2018). Continuous delivery: Building trust in a large-scale, complex government organization. *IEEE Software*, 35(2):38–43. Cited on pp. 2, 3, 9, 17, and 59.

Smeds, J., Nybom, K., and Porres, I. (2015). Devops: A definition and perceived adoption impediments. In Lassenius, C., Dingsøyr, T., and Paasivaara, M., editors, *Agile Processes in Software Engineering and Extreme Programming*, pages 166–177, Cham. Springer International Publishing. Cited on pp. 13 and 14.

Springer, O. and Miler, J. (2022). A comprehensive overview of software product management challenges. *Empirical Software Engineering*, 27(5):106. Cited on pp. xii, 2, 3, 4, 9, 11, 12, 25, 28, 41, 47, 67, and 68.

Springer, O., Miler, J., and Wróbel, M. (2023). Strategies for dealing with software product management challenges. *IEEE Access*, 11:55797–55813. Cited on pp. 11, 25, and 67.

Tanaka, M. L. and Lopez, O. (2023). Outlook on industry-academia-government collaborations impacting medical device innovation. *Journal of Engineering and Science in Medical Diagnostics and Therapy*, 7(2):025001. Cited on p. 3.

Wen, M., Meirelles, P., Siqueira, R., and Kon, F. (2018). *FLOSS Project Management in Government-Academia Collaboration*, pages 15–25. Springer International Publishing, Cham. Cited on pp. 3, 4, 17, 18, 25, and 67.

Wen, M., Siqueira, R., Lago, N., Camarinha, D., Terceiro, A., Kon, F., and Meirelles, P. (2020). Leading successful government-academia collaborations using floss and agile values. *Journal of Systems and Software*, 164:110548. Cited on pp. 1, 3, 17, 18, 25, 59, and 67.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in Software Engineering*. Springer, Berlin, Heidelberg, 1 edition. Cited on pp. 4, 23, 24, and 25.

Yin, R. K. (2014). *Case Study Research: Design and Methods*. SAGE Publications, Thousand Oaks, CA, 5 edition. Cited on p. 25.

Yin, R. K. (2018). *Case Study Research and Applications: Design and Methods*. SAGE Publications, Thousand Oaks, CA, 6 edition. Cited on pp. 4, 23, 24, 25, 26, and 27.