Journal of Software Engineering Research and Development, 2025, 13:5, doi: 10.5753/jserd.2025.3807

A Preliminary Panoramic View of Continuous Software Engineering Adoption in Brazilian Organizations

Paulo Sérgio dos Santos Júnior [**Federal Institute of Education, Science and Technology of Espírito Santo** | ***paulo.junior@ifes.edu.br***]
Fabiano B. Ruy [**Federal Institute of Education, Science and Technology of Espírito Santo** | ***fabianoruy@ifes.edu.br***]
Moisés S. Omêna [**Federal Institute of Education, Science and Technology of Espírito Santo** | ***omenam@ifes.edu.br***]
Monalessa P. Barcellos [**Federal University of Espírito Santo** | ***monalessa@inf.ufes.br***]

Abstract *Context*: Software organizations have faced several challenges, such as the need for faster deliveries, frequent changes in requirements, lower tolerance to failures, and the need to adapt to contemporary business models. Agile practices have allowed organizations to shorten development cycles and increase customer collaboration. However, this has not been enough. Organizations should evolve to continuous and data-driven development in a continuous software engineering approach. Continuous Software Engineering (CSE) consists of a set of practices and tools that support a holistic view of software development with the purpose of making it faster, iterative, integrated, continuous, and aligned with business. Implementing CSE requires changes in the organization's culture, practices and structure, which may not be easy. *Objective*: We aim to provide a preliminary picture of CSE adoption in Brazilian organizations. *Method*: We adapted and used Zeppelin, a diagnostic instrument of CSE adoption based on the Stairway to Heaven Model (StH), to perform a survey with 28 Brazilian organizations aiming at investigating the adoption of CSE practices. After conducting the survey, we interviewed five of the participants to complement the obtained results and better understand how CSE has been performed in their organizations. *Results*: The survey results indicate that organizations have better addressed agile and continuous deployment practices compared to continuous integration and continuous experimentation practices. However, this scenario varies slightly depending on the type of organization. They also show that CSE adoption has been heterogeneous, but there are patterns in the adoption of some practices. The interview results showed that although the five interviewed organizations have distinct CSE scenarios and perform different practices, some practices are common in all of them (particularly agile practices) and some difficulties are faced by all of them (e.g., automated tests). *Conclusion*: Although the StH model proposes a sequential and evolutionary path for CSE adoption, organizations have not always followed it systematically. There are indeed CSE practices that depend on others and thus contribute to sequential implementation. However, organizations tend to adopt the practices gradually, covering different stages, and evolving according to the organization's needs. Moreover, different organizations perform CSE practices in different ways.

Keywords: *Continuous Software Engineering, Stairway to Heaven, Survey, Diagnostic Instrument*

1 Introduction

In the last years, it has been noticed that for producing products that properly meet customers' needs, making well-informed decisions, and identifying business opportunities, new practices should be combined with agile development to enable continuous actions of planning, building, operation, and evaluation (Fitzgerald and Stol, 2017; Barcellos, 2020). Hence, organizations should evolve to continuous and data-driven development in a continuous software engineering approach (Barcellos, 2020; Bosch, 2014).

Continuous Software Engineering (CSE) aims to establish a continuous flow between software-related activities, taking the entire software life cycle into account. It understands the development process not as a sequence of discrete activities, performed by distinct and disconnected teams, but as a continuous flow, considering the entire software life cycle. CSE seeks to transform discrete development practices into more iterative, flexible, continuous alternatives, and keep the goal of building and delivering quality products according to established time and costs (Fitzgerald and Stol, 2017). For that, CSE involves agile practices and goes beyond them by emphasizing the need for continuity, alignment to business, and a broader view of software development (Barcellos, 2020).

In this context, some initiatives have emerged aiming to speed up the development process and improve the connection between its activities. For example, Continuous Integration (Beck, 2000) seeks to eliminate discontinuities between development and delivery. In a similar approach, DevOps (Debois et al., 2011) recognizes that the integration between software development and system operation must be continuous. Extending the need for integration to other levels, BizDev (Fitzgerald and Stol, 2017) advocates that continuity should exist not only in the software process context, but also between software and strategic processes of the organization.

CSE is a recent topic and there are still doubts about how to implement it. Some works have addressed CSE and provided an overview of CSE stages, processes, activities, and practices (e.g., (Olsson et al., 2012) (Johanssen et al., 2018) (Fitzgerald and Stol, 2017) (Barcellos, 2020)). Although these works provide useful knowledge, they are not enough for organizations to identify which practices they should per-

form to implement a certain process or stage and how the organization can evolve in the CSE journey. Considering that, in (Santos Jr et al., 2021) we proposed a diagnostic instrument called *Zeppelin*[1] to aid organizations in identifying their position in the StH stages and planning a path to be followed to achieve continuous and data-driven development. *Zeppelin* helps identify the adoption degree of CSE practices at each StH stage. In this way, organizations can have a panoramic view of the CSE practices they perform, identify where they are in the CSE evolutionary path, and which practices should be improved. *Zeppelin* uses StH (Olsson et al., 2012) as a reference model and also considers Continuous* activities defined in (Fitzgerald and Stol, 2017), CSE elements and categories provided in the Eye of CSE (Johanssen et al., 2018) and CSE processes constituting the CSE framework proposed in (Barcellos, 2020).

In this work, we adapted and used *Zeppelin* to investigate the adoption of CSE practices in Brazilian organizations. We performed a survey with 28 organizations aiming to understand how CSE practices have been adopted and how these organizations are positioned in the StH stages and Eye of CSE categories. The results provided a preliminary panoramic picture of CSE adoption in Brazilian organizations. After the survey, we interviewed five of the participants aiming to get further information about how CSE practices have been performed and complement the survey results.

In summary, the survey results revealed that organizations have better addressed agile [2] and continuous deployment practices than the ones related to continuous integration and continuous experimentation. We also noticed that CSE adoption depends on the organization type. For example, startups tend to focus more on continuous delivery than organizations that develop software for themselves (e.g., for other organizational units of the organization). The results also showed that organizations do not always follow the StH evolutionary path systematically. Due to the agile and flexible software development environment, organizations may perform practices from different stages while evolving from traditional to continuous and data-driven development. Additionally, the interviews revealed that, even in organizations with different CSE scenarios, some practices remain common (e.g., agile practices). Similarly, the same challenges are faced by different organizations (e.g., implementing automated tests). Moreover, although there are commonalities, different organizations perform the same practices in different ways, according to the organization's characteristics, such as culture, team, business goals, and products.

We did not find another study presenting a panoramic view of CSE adoption in Brazilian organizations. This work brings contributions to researchers and practitioners. Based on the preliminary panoramic view of CSE adoption in Brazilian organizations, researchers can identify gaps to be addressed in future research (e.g., develop tools, methods, and guidelines to support relevant practices that organizations have not been able to perform). The preliminary panorama can also be considered to plan future research to support the improvement and advancement of CSE in Brazilian organizations (e.g., investigate why organizations have not adopted certain practices). Practitioners, in turn, can obtain knowledge of CSE practices and apply it to perform CSE. Moreover, they can evaluate their own organizations to get to know how they have applied CSE and which practices can be further addressed.

This paper is part of a larger research project that aims to investigate and provide theoretical and practical advances in CSE. The need to help organizations understand their scenario concerning CSE practices and help them implement or improve such practices was raised in (Barcellos, 2020). Later, Santos Jr et al. (2021) proposed *Zeppelin* and evaluated it in five organizations. After that, Santos Jr et al. (2022a) used *Zeppelin* to investigate CSE in Brazilian organizations. This paper extends the last one mainly by presenting additional views and interpretation of data gathered in the survey and the results from interviews performed with five participants of the survey aiming to understand how CSE has been performed in their organizations. By adding the interviews to the paper, we complement the survey results and provide additional (preliminary) evidence of how CSE practices have been performed in Brazilian organizations. Moreover, we complement the quantitative perspective of survey data with the qualitative perspective of information gathered from interviews.

The paper is organized as follows: Section 2 presents the theoretical background for the paper by addressing CSE. Section 3 discusses related work; Section 4 regards the research design; Section 5 presents ans discuss the survey results; Section 6 concerns the interviews; Section 7 presents some lessons learned; Section 8 discusses threats to validity; and, finally, Section 9 presents some final considerations and future work.

2 Background

CSE involves practices and tools that aim at establishing an end-to-end flow between customer demands and the fast delivery of a product or service (Fitzgerald and Stol, 2017). According to Johanssen et al. (2018), in CSE, customers are proactive, and users and other stakeholders are involved in the process, learning from usage data and feedback. Planning is continuous, so as requirements engineering, which focuses on features, modularized architecture and design, and fast realization of changes. Agile practices are employed, including short development cycles, continuous integration of work, continuous delivery, and continuous deployment of releases. It includes version control of code, branching strategies, fast commit, code coverage, and code reviews. Quality assurance involves automated tests, regular builds, pull requests, audits, and run-time adaption. Knowledge is shared and continuous learning happens, capturing decisions and rationale.

As we said in the Introduction, in the last years, some works have addressed CSE processes and practices. Olsson et al. (2012) defined the *StH Model*, which describes a five-

[1]The name *Zeppelin* was chosen because the diagnostic instrument allows viewing an organization in a panoramic way, as if we were in a zeppelin seeing a city. Besides, Led Zeppelin band created the Stairway to Heaven song (Santos Jr et al., 2021).

[2]Given that we used *Zeppelin* as the instrument in this study, here agile practices refer to the practices included in the Agile Organization stage in *Zeppelin* (this will be further discussed in the paper).

stage evolution path organizations follow to successfully move from traditional to customer data-driven software development. In summary, organizations evolving from *Traditional Development* start by experimenting with one or a few agile teams. Once these teams are successful, agile practices are adopted by the organization, turning it into an *Agile Organization* (AO). As the organization starts showing the benefits of working agile, system integration and verification become involved and continuous integration is adopted. Once *Continuous Integration* (CI) runs internally, lead customers often express an interest in receiving software functionality earlier than through the normal release cycle. They want *Continuous Deployment* (CD) of software. The final stage is Research and Development (R&D) as an Experiment System (RD), when the organization collects data from its customers and uses the installed customer base to run frequent feature experiments to support customer data-driven software development. Figure 1 presents an overview of the StH stages.

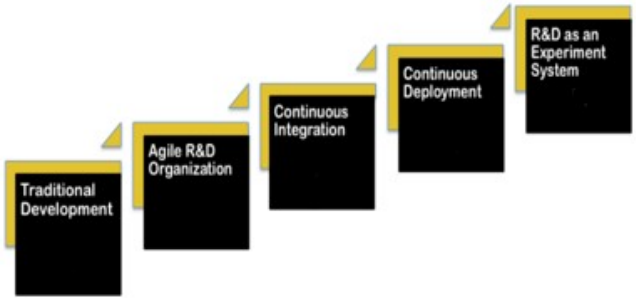

Figure 1. StH Stages.

From interviews performed with CSE practitioners, Johanssen et al. (2018) defined the *Eye of CSE*, which consists of 33 elements (e.g., practices) organized in nine categories. According to the authors, the Eye of CSE can serve as a checklist for practitioners to tackle the subject of CSE by incrementally applying CSE elements and keeping an eye on potential next steps. Figure 2 depicts the *Eye of CSE*.

Fitzgerald and Stol (2017) argue that continuous activities go beyond software engineering activities. They introduce the *Continuous** term, as a set of activities from business, development, operations, and innovation that provides a holistic view of the software life cycle. Continuous planning, continuous security, continuous use, continuous trust, and continuous experimentation are some of the considered Continuous* activities. Figure 3 illustrates the Continuous * activities.

Finally, Barcellos (2020) proposes the *Framework of CSE* (FCSE), which comprises ten processes to be performed in the CSE context (e.g., agile development, continuous integration, continuous deployment, continuous software measurement, continuous knowledge management, and others) and the main relations (information flows and data flows) between them. Activities suggested in (Fitzgerald and Stol, 2017), elements from the Eye of CSE (Johanssen et al., 2018) and StH stages were considered to define the framework. Different from StH, FCSE considers that processes can be performed simultaneously and gradually. Figure 4 provides an overview of the framework proposed by Barcellos (2020).

3 Related Work

Some studies have investigated the use of CSE practices in organizations. For example, Leite et al. (2019) performed a survey to understand the use of DevOps in software organizations. As a result, the authors identified challenges (e.g., building cross-functional team, preserving collaborating departments) and pillars (e.g., human collaboration across departments, automation) to implement DevOps.

We consider the study by Karvonen et al. (2015) the one most similar to ours. The authors used the StH Model and performed a multiple-case study in five Finnish software development organizations to understand how continuous development was implemented. The results showed that organizations adopted CSE practices at different levels and there was a predominance of practices performed at the project/product level. As in our study, Karvonen et al. (2015) also used StH as a reference model and evaluated at which level CSE practices were adopted. However, different from (Karvonen et al., 2015), which involves case studies in five Finnish organizations, our study consists of a survey with 28 organizations and aims to provide a preliminary panorama of CSE adoption in Brazilian organizations. Moreover, our study was performed by using *Zepellin* (Santos Jr et al., 2021), which is based on StH and other works ((Fitzgerald and Stol, 2017) (Johanssen et al., 2018) (Barcellos, 2020)) not considered in (Karvonen et al., 2015).

There is also the work by Leite et al. (2023), which describes how DevOps and Continuous Delivery practices impacted the enterprise architecture of 54 software organizations from different countries (Brazil, Canada, USA, and Western Europe). The results showed that software organizations organize their development and infrastructure workforce in different organizational structures (Segregated Department, Collaborative Department, API mediated department, and single department) based on the applied DevOps and Continuous Delivery practices. Although the study performed by Leite et al. (2023) also investigates CSE aspects in several organizations, the focus is different from our study, which aimed to understand CSE adoption in Brazilian organizations.

4 Research Design

Aiming to investigate the adoption of CSE practices in Brazilian organizations and to get a preliminary panoramic view of CSE adoption in Brazil, we performed a survey. A survey aims at identifying the characteristics of a broad population by generalizing on the data collected from a representative sample of individuals (Easterbrook et al., 2008). Surveys are conducted to produce a snapshot of the situation to capture the current status (Wohlin et al., 2012). We chose this method because, as we aimed at a panoramic view, we needed to reach several organizations and ask about many practices. Hence, carrying out case studies, for example, would be unfeasible.

This section presents information about the research design. It first introduces *Zeppelin*, the instrument used in the study (Section 4.1), then discusses how *Zeppelin* was adapted

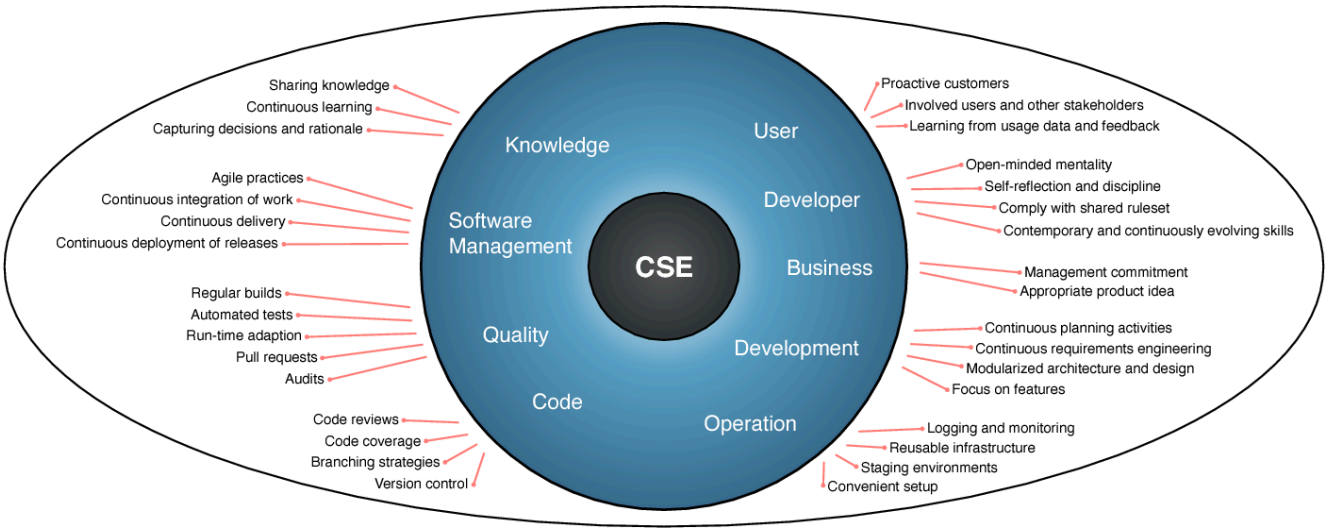

Figure 2. Eye of CSE.

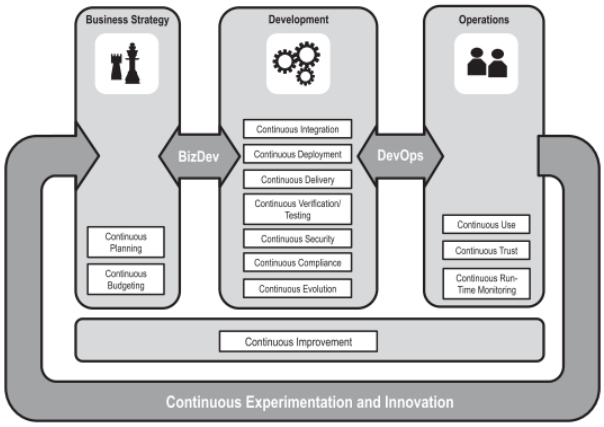

Figure 3. Continuous *.

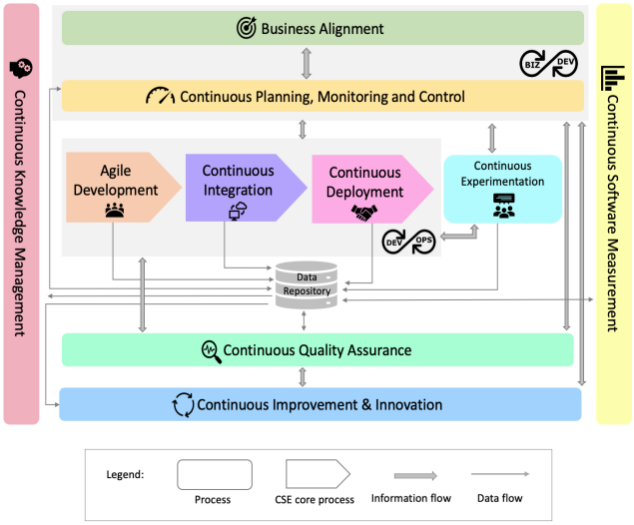

Figure 4. Framework of CSE.

to be used in the survey (Section 4.2), and, lastly, presents the study protocol (Section 4.3).

4.1 Zeppelin: A Diagnostic Instrument for CSE Adoption

Identifying the CSE practices an organization performs and helping it advance in the CSE evolutionary path is a complex and costly activity that involves understanding the organization's culture and analyzing artifacts, processes, tools, people, and other elements present in software development (Santos Jr et al., 2020). Aiming to support organizations to get a panoramic view of how far they have evolved CSE practices and help them identify areas that should be addressed in improvement actions to implement CSE, Santos Jr et al. (2021) proposed *Zeppelin*. It is a diagnostic instrument made up of two components: a *Diagnostic Questionnaire*, which identifies the CSE practices an organization performs and their adoption degree; and an *Analytic Report*, which presents consolidated data from the questionnaire answers, showing a panoramic view of the organization from the CSE perspective and pointing out possible improvement areas.

The *Diagnostic Questionnaire* contains questions to characterize the organization profile and presents 76 statements expressing CSE practices organized in four stages of the StH model: *Agile Organization (AO)* (26 practices), *Continuous Integration (CI)* (15 practices), *Continuous Deployment (CD)* (17 practices) and *R&D as an Experiment System (RD)* (13 practices). In a nutshell, *Zeppelin* considers four stages that include statements referring to practices.

It is important to emphasize that *Zeppelin* is structured based on the StH model. Thus, agile-related practices are not all concentrated in the *Agile Organization* stage. For example, practices related to CI are placed in the CI stage. Therefore, in this study, "agile practices" refer to the practices contained in the *Agile Organization* stage. Moreover, like in the StH model, in *Zeppelin*, CI and CD refer to several practices (e.g., CI involves code integration and other practices related to the CI context). Hence, in *Zeppelin* and, consequently, in this study, CI and CD are treated as stages (involving several related practices) instead of single practices.

In *Zeppelin*, the CSE practices are also categorized considering the Eye of CSE (Johanssen et al., 2018) dimensions (categories and elements). The practices were identified based on the literature (mainly on (Barcellos, 2020), (Fitzgerald and Stol, 2017), (Johanssen et al., 2018), and

(Olsson et al., 2012)) and on the authors' practical experiences.

The questionnaire is used to evaluate which practices have been adopted in the organization and to understand how comprehensive their adoption has been. When applying *Zeppelin*, for each practice, the user must indicate the level at which it is adopted in the organization. The adoption levels were defined based on (Olsson et al., 2012) and are used to capture the comprehensiveness of each practice in the organization and help monitor its evolution. The *Not Adopted* level is used to identify practices that the organization has never used. The *Abandoned* level refers to practices that were discontinued. The *Project/Product* level is used to identify practices not formalized in the organization and used only in a particular project or product. The *Process* level indicates that the practice is formally defined (e.g., by means of procedures, guidelines, business processes, policies) but the team can decide whether to apply it in a project. Finally, a CSE practice is said to be performed at the *Institutionalized* level when it is formally defined and used in all projects.

The adoption degree of each stage (AD) is represented as a percentage and is established by calculating the weighted average of the adoption level (AL) of all practices of that stage (i.e., practices 1 to n, where n is the number of practices related to the stage). Thus, $AD_{stage} = (weight \times AL_{practice1} + ... + weight \times AL_{practicen})/n) \times 100)$. The weights of the adoption levels vary from 0 (zero) (referring to the *Not Adopted level*) to 1.0 (referring to the *Institutionalized level*). Figure 5 presents a fragment of Zeppelin's Diagnosis Questionnaire (Santos Jr et al., 2021).

The *Analytic Report* consolidates data from the answers provided in the *Diagnostic Questionnaire* and presents a panoramic view (by using tables, charts, and text) of CSE practices adoption in the organization by considering three different perspectives: (i) per StH stage (Olsson et al., 2012) (i.e., Agile Organization, Continuous Integration, Continuous Deployment, and R&D as an Experiment System); (ii) per Eye of CSE (Johanssen et al., 2018) dimension (i.e., Deployment, Quality, Software Management, Team, Technical Solution, Knowledge, Operation, Business, and User/Customer), and (iii) per Eye of CSE (Johanssen et al., 2018) element (33 elements, e.g., Agile Practice, Automated Tests, Continuous Deployment Releases, and Continuous Learning). Each CSE element is related to one CSE dimension. By analyzing the different perspectives, the organization identifies its strengths and weaknesses and can define improvement actions accordingly. Figure 6 illustrates some pieces of information contained in the *Analytic Report*.

4.2 Adapting *Zeppelin*

We needed to adapt the version available in (Santos Jr et al., 2021) to make it feasible to apply it in the study. The *Zeppelin* version proposed in (Santos Jr et al., 2021) is available as an electronic spreadsheet. To automatize data collection, we turned it into a form using Google Forms. The form contains a consent term, in which participants declare to accept to participate in the study, and seven sessions: *Organization*, with questions to characterize the organization (e.g., organization type and size); *User*, to characterize the person answering the questionnaire on the organization's behalf (e.g., position, knowledge and experience in CSE practices); and four sessions concerning StH stages. We rewrote some statements and rearranged the order in which they appear in the questionnaire to make them clearer and improve user experience when answering it. We also turned some sentences into one. As a result, we reduced from 76 to 71 statements (26 referring to AO, 15 to CI, 17 to CD, and 13 to RD).

We also created a simpler version of the *Analytic Report*. The report proposed in (Santos Jr et al., 2021) is produced as a result of the evaluation of a single organization. It is detailed and depends on human intervention. Considering that in this study we needed to reach several organizations, it would be unfeasible to produce a detailed report to each of them. Hence, we created a simpler automatic report that summarizes the information provided in the questionnaire. Table 1 presents some statements in the Diagnostic Questionnaire of the Zeppelin version used in the survey.

4.3 Study Protocol

Aligned with the study goal, we defined the following main **research question**: *(RQ1) How has CSE been adopted in Brazilian software organizations?*. This question has the purpose of giving a general view of CSE adoption in Brazilian organizations. In this study, StH stages and Eye of CSE categories are used to provide the general view considering different perspectives. We considered these perspectives because they are addressed by *Zeppelin* (Santos Jr et al., 2021).

From this question, we defined other two questions to complement the general panorama by investigating some specific aspects that are not directly addressed by Zeppelin, but that can be explored from data collected using its questionnaire and are helpful to understand CSE adoption: *(RQ2) How have different types of organization adopted CSE?*, *(RQ3) Which repeatable behaviors have happened in CSE adoption?*. With RQ2 we intend to investigate differences in CSE adoption due to the organization type. With RQ3 we seek to identify possible patterns of CSE adoption (e.g., correlated practices that tend to be all adopted or all not adopted).

The **instrument** used in the study was *Zeppelin*, which was adapted as described in Section 4.2.

The **procedure** followed in the study consisted of seven steps. In the first step, we ran a small pilot to evaluate the form and the study protocol. We asked two software engineers with experience in CSE to answer the questionnaire and report problems, suggestions, and response time. Based on the provided feedback, we made minor adjustments. In the second step, we sent messages inviting people from different organizations to participate in the study. The invitation was sent via social networks (LinkedIn, WhatsApp, and Instagram) and email. Considering that the questionnaire was quite long, it was possible that people were not willing to answer it. Thus, we contacted some researchers and practitioners from our contact network (including people from the five Brazilian regions) and asked them to participate in the study or indicate organizations that we could invite. The third step consisted of gathering data from the answered questionnaires, representing them in tables and graphs, and analyzing them. In the fourth step, we sent the analytic report to each

Continuous Integration

Implementation of concepts and techniques related to code integration and automated testing, such as TDD, automated build and testing, and test environments.

#	Statement	Adoption Level	Comments *(provide details on how the practices are performed or, if don't, what the organization*
CI.01	The software architecture is modular in order to allow automated testing.		
CI.02	The software architecture is modular in order to allow automated builds.	Not Adopted	
CI.03	Code is constantly and automatically integrated.	Abandoned	
CI.04	Tests are automatically executed periodically (e.g., whenever new code is integrated), in a` testing environment, to verify code quality (e.g., coverage, correctness).	Project / Product Process Institutionalized	
CI.05	Automated tests are used to assess whether the implemented software meets established requirements.		
CI.06	Builds occur frequently and automatically.		
CI.07	Builds are canceled if one or more tests fail		
CI.08	There is version control of software artifacts (e.g., code, tests, scripts, etc.) in a repository.		
CI.09	Good check-in practices are applied in the development trunk (e.g., use of tools like GitFlow).		
CI.10	There are practices that allow organizations or individuals external to the project to contribute to the product implementation (i.e., produce and integrate code into the product being developed)		
CI.11	Data is collected for metrics that allow evaluation of the continuous integration process (e.g.,number of canceled builds, number of code integrations performed).		
CI.12	Data produced in continuous integration environments (e.g., build dates, number of tests executed, and percentage of coverage) are stored in one (or more) data repository.		
CI.13	The continuous integration process (including automated testing) is continuously evaluated and improved.		Inform the adopted practices:
CI.14	Data stored in the data repository(ies) are used to improve the product and the continuous integration process.		Inform some used metrics:
CI.15	The organization adopts practices for sharing knowledge related to continuous integration (e.g., internal lectures, tutorials, knowledge repositories, guild implementations).		

Figure 5. Fragment of the Diagnosis Questionnaire with practices related to Continuous Integration (Santos Jr et al., 2021).

Dimension	Degree of Adoption				
	Stages of StH				Organization
	AO	CI	CD	IS	
Development	69%	58%	75%		67%
Quality	50%	65%	75%		64%
Sw Management	73%	63%	69%		70%
Team	88%			75%	81%
Technical Solution	50%	75%			67%
Knowledge	50%	50%	50%	58%	53%
Operation				88%	88%
Business	100%		60%	92%	75%
User/Customer			94%	70%	84%
Degree of Adoption	**69%**	**64%**	**70%**	**76%**	**69%**

Adoption Level	Number of CSE Practices				
	Stages of StH				Organization
	AO	CI	CD	IS	
Not Adopted			2	1	3
Abandoned				1	1
Project/Product	11	9	3	1	24
Process	10	8	9	3	30
Institutionalized	5	1	5	7	18
Degree of Adoption	**69%**	**64%**	**70%**	**76%**	**69%**

Number of Adopted Practices		
Adoption Level	**Practices**	**(%)**
Not Adopted	3	4%
Abandoned	1	1%
Project / Product	24	32%
Process	30	39%
Institutionalized	18	24%
Adopted Practices	**72**	**95%**
Total	**76**	**100%**

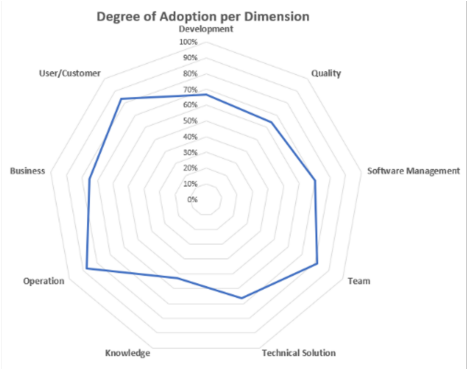

Figure 6. Fragments of the Analytic Report (Santos Jr et al., 2021).

Table 1. Examples of statements related to StH stages defined in the Diagnostic Questionnaire of Zeppelin

StH Stage	Statement	
Agile Organization	AO.04	Requirements are defined and prioritized according to customer needs and are periodically reviewed.
	AO.05	The scope of the project is defined gradually, using the Product Backlog (or an equivalent artifact).
	AO.08	The development process is agile, being carried out iteratively, in short cycles, in which product requirements defined in a Product Backlog (or equivalent artifact) are selected, recorded in a Sprint Backlog (or equivalent artifact), and developed.
	AO.15	Frequently the team meets and reflects on development progress within the scope of what was defined for the current time box and adjusts tasks if necessary.
	AO.17	The team meets frequently throughout the project to discuss improvements in team member skills (e.g., in retrospective meetings).
	AO.20	Good testing practices (e.g., automated testing) are adopted.
	AO.21	Data is collected for metrics that allow evaluating aspects of the quality of the produced artifacts and the product (e.g., cyclomatic complexity).
	AO.22	Data is collected for metrics that allow evaluating performance aspects of the agile development process (e.g., work in progress, velocity).
Continuous Integration	CI.02	The software architecture is modular to allow automated builds.
	CI.03	Code is integrated constantly and automatically.
	CI.06	Builds occur frequently and automatically.
	CI.08	There is version control of software artifacts (e.g., code) in a repository.
	CI.10	There are practices that allow organizations or people external to the project to act on the implementation of the product.
	CI.11	Data is collected for metrics that allow evaluating the continuous integration process (e.g., number of canceled builds, number of performed code integration).
	CI.14	Data stored in the data repository(ies) is used to improve the product and the continuous integration process.
Continuous Deployment	CD.03	There is a continuous deployment process by releases.
	CD.07	Customers can test the product as soon as new features are deployed.
	CD.08	The organization's business model is constantly evaluated and revised (when necessary) based on information from customers/consumers.
	CD.09	Marketing strategies are constantly evaluated and revised (when necessary) based on information from lead customers (clients/consumers most relevant to the organization).
	CD.10	Sales strategies are constantly evaluated and revised (when necessary) based on information from lead customers.
	CD.11	Alignment between product development and the organization's business is maintained through continuous checks, in short cycles.
	CD.13	Data is collected for metrics that allow evaluating the continuous delivery process.
	CD.16	Data stored in the data repository(ies) is used to improve the product and continuous delivery process.
	CD.17	Actions are carried out to share knowledge related to continuous delivery (e.g., internal lectures, tutorials, knowledge repositories, implementation of guilds).
R&D as an Experiment System	RD.05	Experiments (e.g., A/B tests) are conducted with customers/consumers to improve products.
	RD.06	Technologies are adopted (e.g., cloud technologies) to allow experimentation to be enhanced.
	RD.08	There is an information flow between the strategic level and the development area, allowing customer/consumer data to be used in an aligned way in making technical and business decisions.
	RD.09	Data from the customer/consumer data repository is used in decision-making by the software development area.
	RD.10	Data from customer/consumer data repositories are used in decision-making by the business area.
	RD.12	The process of continuous experimentation is continually evaluated and improved.

study participant. In the fifth step, we invited the survey participants for an interview to better understand how CSE practices have been applied in the respective organizations. We sent the invitations by email. In the sixth, we interviewed the participants who accepted our invitation. In the seventh and last step, we analyzed information obtained from the interviews. The form used in the study is in the study package available in (Santos Jr et al., 2022b).

The **participants** of the study were people with knowledge of and experience in CSE (but not necessarily experts) and who work in Brazilian organizations that perform CSE practices.

5 Using *Zeppelin* to Investigate CSE Adoption

This section presents the survey and its main results. Section 5.1 concerns the survey execution and summarizes the results, while Section 5.2 discusses the results

5.1 Survey Execution and Results

The *Zeppelin* questionnaire was made available in March 2022 and data was collected until late April 2022. The study involved 28 participants from different Brazilian organizations. All the received forms were completely answered. We wanted to get one participant by organization in the study. When contacting people from our network, we informed them that we needed one person to participate in representing the organization. However, as we invited people by using different channels, we could not avoid people from the same organization answering the questionnaire. Even so, all the answers we received were from participants representing different organizations. In fact, two participants were from the same company (the same company "name") but they worked in different branches, located in different states, and that work independently and in different ways. Thus, for the study purposes, we considered them as different organizations.

The organizations' profile was identified through questions regarding its type, number of employees, and geographic region. Regarding types, 43% are *Organizations with an IT Department* (i.e., organizations that have IT department(s) to produce software and services for supporting the business), 32% are *Startups* (organizations that have a repeatable and scalable business model), and 25% are *Software Houses* (i.e., organizations that develop software for other organizations). Considering size, 7% of the organizations have between 1 and 9 employees, 18% have between 10 and 49 employees, 11% have between 50 and 99 employees, and 64% have more than 99 employees. As for location, most organizations (61%) are in the Southeast region, 11% in the South, 11% in the North, 11% in the Northeast, and 6% in the Midwest, covering the five Brazilian geographic regions. Table 2 summarizes the organizations type, size, and region.

Table 2. Organizations type, size, and region.

Type	**Quantity**	**%**
Organization with an IT Department	12	43%
Software House	7	25%
Startup	9	32%
Size	**Quantity**	**%**
Between 01 and 09 employees	2	7%
Between 10 and 49 employees	5	18%
Between 50 and 99 employees	3	11%
More than 99 employees	18	64%
Region	**Quantity**	**%**
Midwest	2	6%
North	3	11%
Northeast	3	11%
South	3	11%
Southeast	17	61%

The study participants are professionals with knowledge of and experience in CSE. The participants' profile was identified through questions regarding their current job positions, educational level, knowledge of CSE, and practical experience in CSE. Most participants (64%) declared to play roles directly related to software development projects (three participants are Product Owners, two are Scrum Masters, seven are Tech Leaders, three are Project Managers and, three are Developers), while 36% play roles related to business management (three are Managers and seven are Directors). As for education, two participants (7%) have a Ph.D. degree, 11 (40%) have a Master's degree, and 15 (53%) have a Bachelor's degree.

The participants were asked about their knowledge of and practical experience in CSE. Thus, the knowledge and experience levels were declared by the participants. The answers were given to each StH stage. A brief explanation of each stage was provided for the participants, so that they could properly answer the question. Table 3 summarizes the results. Concerning knowledge, *none* means that the participant does not know the subject; *low* means that knowledge was obtained from reading materials or through a short course (up to 4 hours); *medium* means that knowledge was obtained in undergraduate projects or a course lasting more than 4 hours; *high* means that the participant is a specialist in the subject (e.g., is certified by some renowned institution, is author of a master or doctorate research on the subject). Regarding experience, *none* means that the participant has never worked on the subject; *low* refers to up to 1 year of experience on the subject; *medium* refers to 1 to 3 years of experience; and *high* to more than 3 years of experience. In the following, we present a data synthesis for each research question.

Table 3. Participant's knowledge and experience.

Item	**Level**	**AO**	**CI**	**CD**	**RD**
Knowledge	None	0	1	0	5
	Low	1	4	6	6
	Medium	15	12	10	12
	High	12	11	12	5
Experience	None	0	2	1	8
	Low	2	5	5	8
	Medium	7	12	10	7
	High	19	9	12	5

RQ1: ow has CSE been adopted in Brazilian software organizations?

To answer this question, we represented collected data in different ways to provide different and complementary views of CSE adoption. First, we took the adoption degree into account (calculated as explained in Section4.1), to provide a view of CSE adoption based on the level at which the practices are adopted (Product/Project, Process, Institutionalized, Not Adopted, and Abandoned). In this context, we looked at the average adoption degree of CSE practices at each StH stage and CSE category. Second, aiming to investigate the levels at which CSE practices have been adopted and identify the predominant adoption level, we represented the average percentage of adopted CSE practices by level. Last, to understand the adoption of each CSE practice, we represented the average percentage of adoption of each practice at each StH stage. Next, we present some of the charts produced to provide these different views of CSE adoption in organizations.

Figure 7 shows the adoption degree related to each StH stage. The adoption degree was calculated using the procedure established in Zeppelin (see Section 4.1). As can be seen, organizations have better covered agile practices AO (53.3%), followed by CD practices (47.4%), CI (42.7%), and RD (33.2%).

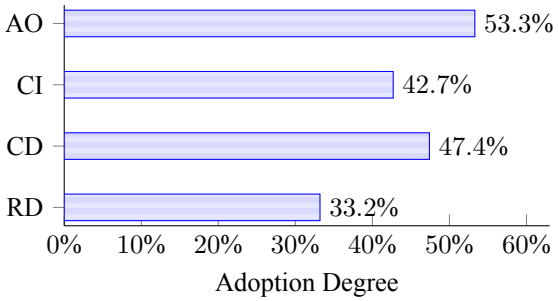

Figure 7. Adoption degree per StH stage.

Figure 8 provides a complementary view to Figure 7 by showing the adoption degree distribution at each stage. In the figure, data from each stage (i.e., the adoption degree of each practice considering the 28 organizations)is represented by a box and a vertical line in the same color. The highest and lowest values in that line represent the maximum and minimum values of the data set (i.e., the highest and lowest adoption degree of a practice at that stage), respectively. The horizontal line inside the box denotes the median, while the x indicates the mean. The box is drawn from the lower quartile (1rst quartile) to the upper quartile (3rd quartile) and, thus, represents where are the 50% of adoption degrees more frequent in the data set. Therefore, the smaller the box and the closer the maximum and minimum values, the less dispersed the distribution of values. In other words, more similar the organizations behave concerning the adoption of CSE practices at that stage. On the other hand, the larger the box size and the distance between maximum and minimum values, the greater the dispersion of the values (i.e. the organizations behave in a more varied way when adopting CSE practices at that stage).

The adoption degree of AO practices ranges from 35% to 70%. The ones with the highest adoption degrees are related to requirements definition and tracking (*(AO.04) Requirements are defined and prioritized according to customer needs and are periodically reviewed* and *(AO.05) The scope of the project is defined gradually, using the Product Backlog (or an equivalent artifact)*) and scope progress *((AO.15) Frequently the team meets and reflects on development progress within the scope of what was defined for the current timebox and adjusts tasks if necessary*), while the ones with the lowest adoption degrees are related to testing (*(AO.20) Good testing practices (e.g., automated testing) are adopted*) and data collection (*(AO.21) Data is collected for metrics that allow evaluating aspects of the quality of the produced artifacts and the product (e.g., cyclomatic complexity)* and *(AO.22) Data is collected for metrics that allow evaluating performance aspects of the agile development process (e.g., work in progress, velocity)*).

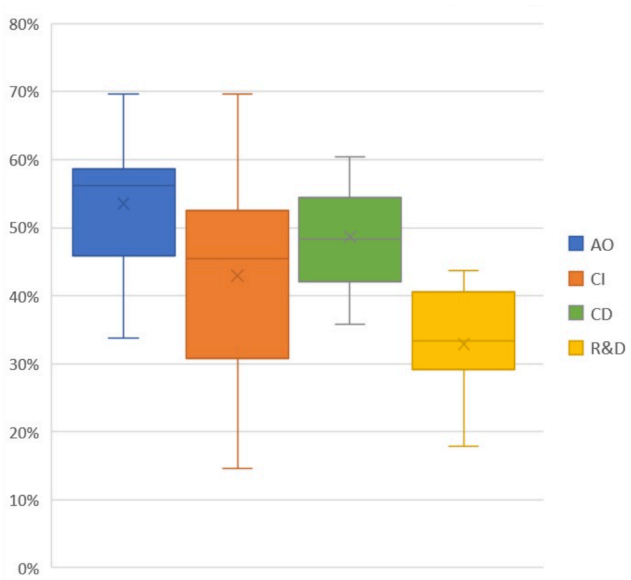

Figure 8. Practices adoption - Distribution per stage.

CI practices have the widest range of adoption degree, going from 15% to 70%. Practices with the highest adoption degrees address architecture (*(CI.02) The software architecture is modular to allow automated builds*) and version control aspects (*(CI.08) There is version control of software artifacts (e.g., code) in a repository*), while the ones with the lowest adoption degrees involve the participation of external stakeholders (*(CI.10) There are practices that allow organizations or people external to the project to act on the implementation of the product*) and the use of data for process improvement (*(CI.11) Data is collected for metrics that allow evaluating the continuous integration process (e.g., number of canceled builds, number of performed code integration)* and *(CI.14) Data stored in the data repository(ies) is used to improve the product and the continuous integration process*).

Concerning CD, the adoption degree varies from 36% to 60%. The DC practices with the highest adoption degrees are related to the alignment between development and business (*(CD.08) The organization's business model is constantly evaluated and revised (when necessary) based on information from customers/consumers* and *(CD.11) Alignment between product development and the organization's business is maintained through continuous checks, in short cycles*) and the delivery of new features (*(CD.03)There is a continuous deployment process by releases* and *(CD.07) Customers can*

test the product as soon as new features are deployed). On the other hand, practices with the lowest adoption degrees regard collecting and using data for product and process improvement (*(CD.13) Data is collected for metrics that allow evaluating the continuous delivery process (e.g., number of releases, density of defects in releases)* and *(CD.16) Data stored in the data repository(ies) is used to improve the product and continuous delivery process)* and knowledge sharing (*(CD.17) Actions are carried out to share knowledge related to continuous delivery (e.g., internal lectures, tutorials, knowledge repositories, implementation of guilds)*).

Last, the adoption degree of RD practices is the lowest and ranges from 18% to 44%. The ones with the highest adoption degrees are related to technologies (*RD.06 - Technologies are adopted (e.g., cloud technologies) to allow experimentation to be enhanced*) and use of user feedback in decision-making (*(RD.08) There is an information flow between the strategic level and the development area, allowing customer/consumer data to be used in an aligned way in making technical and business decisions*). The ones adopted with the lowest degrees focus on conducting A/B experiments (*(RD.05) Experiments (e.g., A/B tests) are conducted with customers/consumers to improve products*) and improving the continuous experimentation process (*(RD.12) The process of continuous experimentation is continually evaluated and improved*).

Figure 9 presents the adoption degree related to the Eye of CSE categories. As it can be observed, the categories with the highest adoption degrees are Technical Solution (58.8%) and Software Management (52,5%), while the categories with the lowest degrees of adoption are Quality (36.6%) and Team (39.2%).

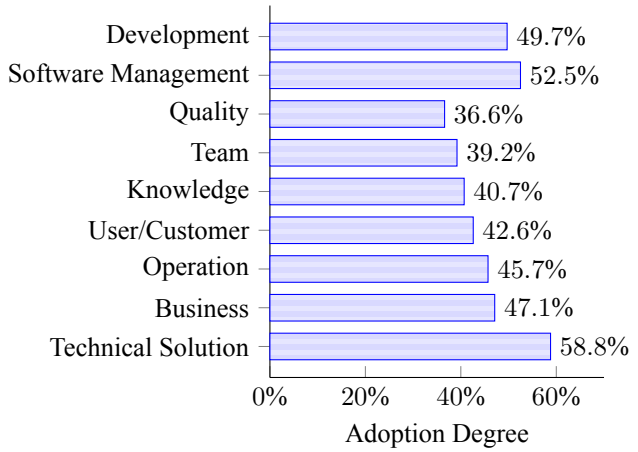

Figure 9. Adoption degree per Eye of CSE category.

Aiming to represent the level at which CSE practices have been adopted, Figure 10 shows the average percentage of CSE practices at each level per StH stage. In AO, CI, and CD there is a predominance of practices at the Project/Product level, while in RD there is a predominance of not adopted practices. In total (i.e., considering all StH stages), the predominance occurs at Project/Product level (35,4%), followed by Institutionalized (24.3%), Not Adopted (18.8%), Process (17.8%), and Abandoned (3.7%).

Figure 11 provides an overview of the practices adoption per stage, regardless of the level at which the practice is

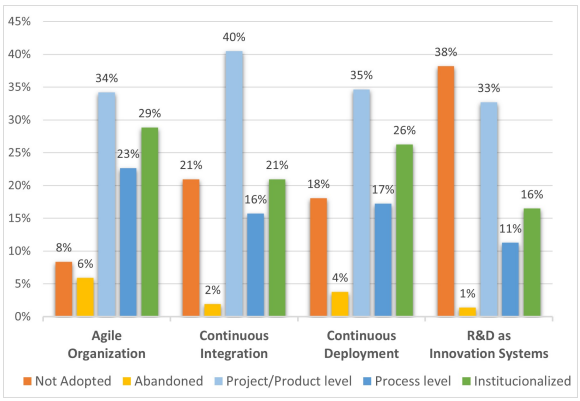

Figure 10. Practices per adoption level.

adopted. Thus, data presented in Figure 11 considers the average percentage of practices adopted at each stage. The results are consistent with the ones based on the adoption degree (i.e., there is a predominance of practices related to AO, followed by CD, CI, and RD). In the figure, each practice is identified by its id in the questionnaire. The practices referred by each id can be found in the study package (Santos Jr et al., 2022b).

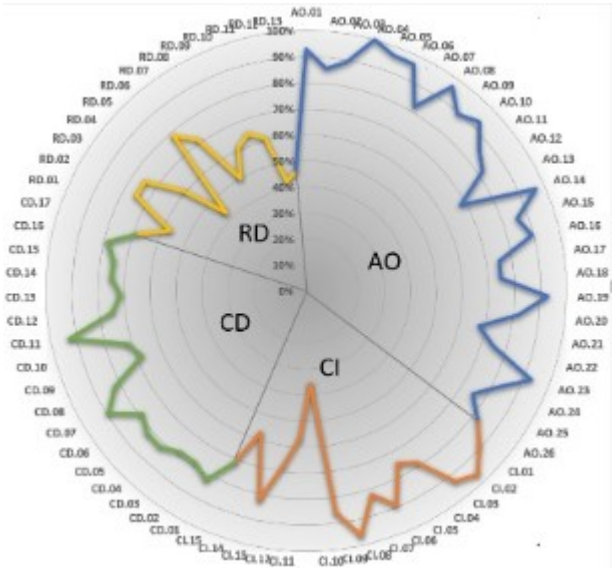

Figure 11. CSE practices adoption.

To provide a view of practices adoption in each organization, Figure 12 presents the number of practices adopted by each participant Organization at each StH stage.

By considering the practices individually, we identified the ones most and least adopted by the organizations. For that, firstly, we identified the number of organizations adopting each practice at any level of adoption (Qty). After that, we calculated the percentage of organizations that adopted the practice (Adpt) and, finally, we calculated the average degree of adoption of each practice (Average Degree). Table 4 and Table 5 present the Top 3 CSE practices of each stage with the highest and lowest average degrees of adoption.

RQ2: How have different types of organization adopted CSE?

To obtain data to answer this research question, we categorized the collected data according to the organization type.

Table 4. The most adopted CSE Practices.

StH	Practice	Qty	Adpt	Average Degree
AO	AO.04 - Requirements are defined and prioritized according to customer needs and are periodically reviewed	28	100%	68%
	AO.08 - The development process is agile, being carried out iteratively, in short cycles (e.g., two weeks), in which product requirements defined in a Product Backlog (or equivalent artifact) are selected, recorded in a Sprint Backlog (or equivalent artifact) and developed	27	96%	64%
	AO.15 - Frequently, the team meets and reflects on development progress within the scope of what was defined for the current time box and adjusts tasks if necessary (e.g., in daily or stand-up meetings)	24	86%	64%
CI	CI.08 - There is version control of software artifacts (e.g., code) in a repository	27	96%	70%
	CI.02 - The software architecture is modular and allows automated builds	27	96%	58%
	CI.03 - Code is integrated constantly and automatically	26	93%	54%
CD	CD.11 - Alignment between product development and the organization's business is maintained through continuous checkings, in short cycles	26	93%	61%
	CD.07 - Customers can test the product as soon as new features are deployed	25	89%	58%
	CD.08 - The organization's business model is constantly evaluated and revised (when necessary) based on information from customers/consumers	23	82%	58%
RD	RD.06 - Technologies (e.g., cloud technologies) are adopted to enhance experimentation	22	79%	46%
	RD.02 Feedback (data and opinions) from customers/consumers is captured continuously and automatically and is used to improve products (improve existing features and identify new ones)	21	75%	44%
	RD.03 - The organization identifies new business opportunities based on feedback automatically captured from customers/consumers	21	75%	44%

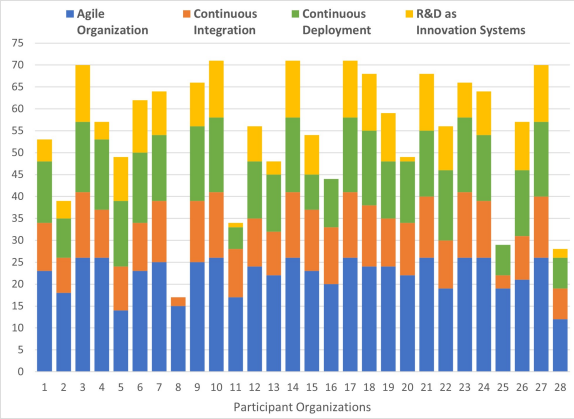

Figure 12. CSE practices adoption per organization.

Table 6 presents the adoption degree at each StH stage per organization type. As can be observed, Organizations with an IT Department have the highest adoption degrees in practices related to AO (48.8%) and CI (44.6%) and the lowest adoption degree occurs in practices related to RD (24.6%). Software Houses got the best rates (65.1% in AO, 48.9% in CI, 55.1% in CD, and 45.8% in RD). Finally, Startups perform better in AO (50.2%) and CD (50.1%) than in CI (35.5%) and RD (34.9%).

Regarding the Eye of CSE categories, all the organization types have the highest adoption degree in Technical Solution. As for the lowest adoption degree, for Organizations with an IT Department it occurs in Team, while for Startups and Software Houses, it occurs in Quality. Table 7 summarizes the adoption degrees by organization type and at each category. Figure 13 illustrates data from Table 7 providing a general view of the adoption degrees per organization type and compared to the total adoption degree (i.e., considering all organization types).

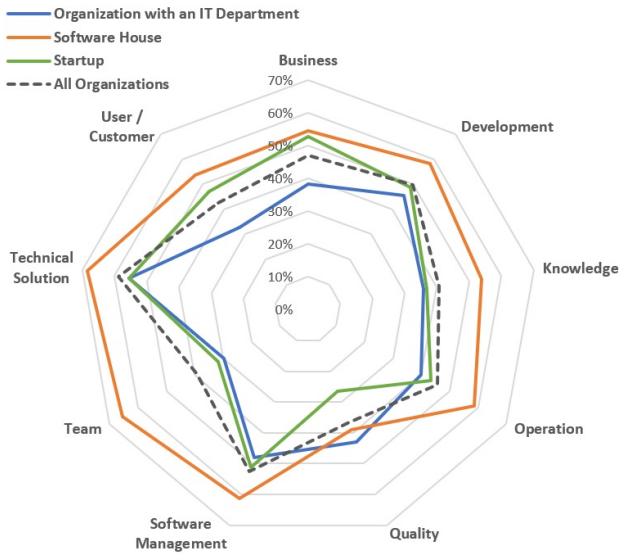

Figure 13. Adoption degree per organization type and StH stage.

RQ3: Which repeatable behaviors have happened in CSE adoption?

Similar to RQ2, for answering this question we need to combine different data provided by the participants. Thus, we applied clustering and Spearman's rank correlation coef-

Table 5. The less adopted CSE Practices.

StH	Practice	Qty	Adpt	Average Degree
AO	AO.21 - Data is collected for metrics that allow evaluating aspects of the quality of the produced artifacts and the product (e.g., cyclomatic complexity)	19	68%	34%
	AO.22 - Data is collected for metrics that allow evaluating performance aspects of the agile development process (e.g., work in progress, velocity)	21	75%	37%
	AO.20 - Good testing practices are adopted (automated testing, test-driven development, etc.).	23	82%	38%
CI	CI.10 - There are practices that allow organizations or people external to the project to act on the implementation of the product	10	36%	16%
	CI.15 - Actions are carried out to share knowledge related to continuous integration (e.g., internal lectures, tutorials, knowledge repositories, implementation of guilds)	16	57%	22%
	CI.11 - Data is collected for metrics that allow evaluating the continuous integration process (e.g., number of canceled builds, number of performed code integration)	16	57%	28%
CD	CD.16 - Data stored in the data repository(ies) is used to improve the product and the continuous delivery process	22	79%	34%
	CD.15 - The continuous delivery process is continually evaluated and improved	21	75%	38%
	CD.13 - Data is collected for metrics that allow evaluating the continuous delivery process	20	71%	40%
RD	RD.05 - Experiments (e.g., A/B tests) are conducted with customers/consumers to improve products	12	43%	16%
	RD.12 - The continuous delivery process is continually evaluated and improved	12	43%	23%
	RD.08 - There is an information flow between the strategic level and the development area, allowing customer/consumer data to be used in an aligned way to make technical and business decisions	14	50%	27%

Table 6. Adoption degree per organization type and StH stage.

StH Stage	Organization with an IT Department	Software House	Startup
AO	48.8%	65.1%	50.2%
CI	44.6%	48.9%	35.5%
CD	40.8%	55.1%	50.1%
RD	24.6%	45.8%	34.9%
All	**41.5%**	**55.7%**	**44.3%**

ficient (Myers and Sirois, 2004) to explore data. We were not able to find relevant findings using clustering methods, because we have only 28 organizations. By applying Spearman's correlation coefficient and considering the practices' adoption degree, we found a correlation between some CSE practices. Here we briefly present three correlations. Others can be found in the study package (Santos Jr et al., 2022b).

Figure 14 shows the correlation between two practices related to CI: *(CI.03) Code is integrated constantly and automatically* and *(CI.06) Builds occur frequently and automatically*. The chart is composed of two components. The first one is a bar chart that represents the adoption degree of the considered CSE practices. The second one is a line chart that represents the correlation between the adoption degree of the two considered CSE practices. For the CSE practices represented in Figure 14, spearman=0.923 and R^2=0.859. It means that it was noticed that organizations that performed build frequently and automatically also integrated code frequently and automatically. On the other hand, organizations that did not perform build frequently and automatically also did not integrate code frequently and automatically. After all, frequent builds are not done without continuous integration. This is an expected result because it is a fundamental aspect of CI/CD.

Table 7. Adoption degree by organization type and Eye of CSE category.

Category	Organization with an IT Department	Software House	Startup	All
Business	38.3%	54.6%	52.8%	**47.1%**
Development	45.5%	58.2%	48.7%	**49.7%**
Knowledge	35.9%	53.8%	37.0%	**40.7%**
Operation	40.0%	58.6%	43.3%	**45.7%**
Quality	42.9%	38.9%	26.4%	**36.6%**
Software Management	48.1%	61.4%	51.3%	**52.5%**
Team	29.6%	65.4%	31.7%	**39.2%**
Technical Solution	55.6%	68.6%	55.6%	**58.8%**
User/Customer	32.6%	53.7%	47.1%	**42.6%**

Concerning Continuous Deployment, it was identified a correlation between *(CD.09) Marketing strategies are constantly evaluated and revised (when necessary) based on information from lead customers* and *(CD.10) Sales strategies are constantly evaluated and revised (when necessary) based on information from lead customers*. Figure 15 represents these practices, with spearman=0.944 and R^2=0.905.

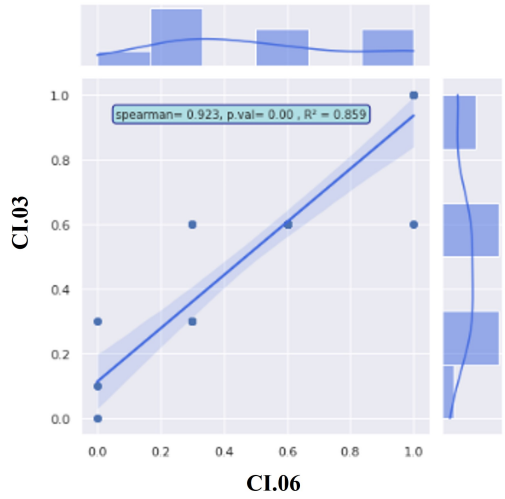

Figure 14. Correlation between CI.03 and CI.06.

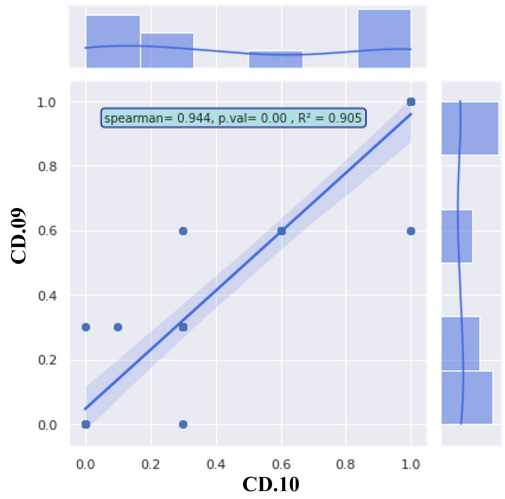

Figure 15. Correlation between CD.09 and CD.10.

As for practices related to Research & Development as an Experiment System, there is a correlation between *(RD.09) Data from the customer/consumer data repository is used in decision making by the software development area* and *(RD.10) Data from the customer/consumer data repository is used in decision making by the business area*, with spearman=0.992 and R^2=0.989, as represented in Figure 16.

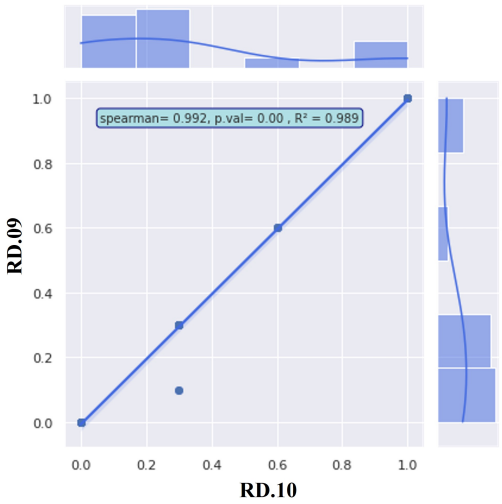

Figure 16. Correlation between RD.09 and RD.10.

5.2 Analysis and Interpretation

Aiming at answering the research questions defined in the study, in this section we analyze and discuss data presented in Section 5.1. Our main goal is to provide an overview of how CSE practices have been adopted in the studied organizations and, thus, obtain a preliminary panoramic picture of CSE adoption.

Concerning *How CSE has been adopted in Brazilian software organizations (RQ1)*, when taking the StH stages as a reference, it is possible to notice that the stage with the highest adoption degree is *Agile Organization* (53.3%) and the one with the lowest adoption degree is *R&D as Innovation System* (33.2%). This was indeed expected because CSE usually starts from agile practices, while *R&D as an Experiment System*, which involves customer data-driven development and experiments with customer data, is only achieved when the organization gets some maturity in CSE and it is able to continuously and automatically collect user feedback and use it to make improvements at software development and business levels (Olsson et al., 2012). Moreover, depending on the organization business, some RD practices (e.g., test A/B) may be less applicable. Surprisingly, *Continuous Integration* had a smaller adoption degree (42.7%) than *Continuous Deployment* (47.6%). This might indicate that some practices related to *Continuous Integration* have been neglected (or not systematically performed), even when *Continuous Deployment* is performed. In this context, we noticed that several organizations have not covered practices related to automatized tests, which can be considered a bad practice because to continuously integrate code, the code must be tested in a proper test environment (Shahin et al., 2017).

When taking the Eye of CSE categories into account, *Technical Solution* and *Software Management* had the highest adoption degrees (58.8% and 52.5% respectively) while *Quality* had the lowest (36.6%). This is somehow consistent with the results referred to the StH stages because the focus has been on the product and management aspects, which are present mainly in agile practices. CSE practices related to *Quality* have received less attention. This reinforces the perspective based on the StH stages, which suggested that some practices related to tests should be improved.

As it was explained in Section 2, by using the adoption degree, *Zeppelin* considers not only if the practices are adopted but also how they are adopted (i.e., occasionally, in some project/product; systematically, through organizational policies/processes that may - or may not - be used, or institutionalized in all organization). Aiming to provide a complementary view, we also analyzed data to identify practices that have been adopted by the organizations, regardless of the adoption level. As a result, we found out that all 71 practices are adopted by the organizations to some extent, with one practice adopted by all organizations (*(AO.04) Requirements are defined and prioritized according to customer needs and are periodically reviewed)*. This can be explained by the fact that this practice addresses requirement elicitation, which is a basic and key activity in software development (whatever the software engineering approach used) and many other development activities depend on it. In contrast, *(CI.10) There*

are practices that allow organizations or people external to the project to act on the implementation of the product was adopted by only ten organizations. This suggests that the organizations have not been concerned with (or interested in) external people participating in implementation tasks. We noticed that the percentage of adopted practices is not homogeneous.

We noticed that considering the top 3 most adopted practices of each stage, practices from *Agile Organization* and *Continuous Integration* stand out as the ones with the highest values. The AO practices most adopted refer to well-known agile practices, including requirements definition and prioritization, iterative development in short cycles, use of backlog, and frequent meetings. The CI practices most adopted address version control, automated builds in modular architecture, and frequent and automated code integration. This indicates a concern with core activities to implement CI. As for CD, the most adopted practices indicate a concern with the alignment between development and operation, and clients testing new features earlier and frequently. Regarding RD, the most adopted practices have been adopted in fewer organizations than the ones from other stages and have focused on technologies and capturing user feedback continuously and automatically to improve products and identify new business opportunities. We observed that these practices refer to fundamental aspects of CSE: agile development, frequent and automated build and code integration, alignment between development and operation, and continuous and automated capture of user feedback and its use for continuous improvement. In contrast, the practices adopted in fewer organizations reveal some aspects that have been neglected, mainly regarding metrics and their use to improve products and processes.

Considering the StH stages, *Agile Organization* has an average adoption of 85.7% of the practices, followed by *Continuous Deployment*, with 78.2%, *Continuous Integration*, with 77.1%, and *R&D as an Experiment System*, with 60.4%. These results are consistent with the ones based on the adoption degree. However, the average of adopted practices is higher than the adoption degree because organizations have adopted CSE practices at different levels. In fact, there has been a predominance of adoption at Project/Product level (35,4%), suggesting that organizations have performed CSE practices, but they have not been institutionalized or systematically performed. 24.3% of the adoption is at the Institutionalized level and 17.8% at the Process level. This may suggest that organizations still need to mature some CSE practices. The predominance of practices at Project/Product level is consistent with the results of the study by Karvonen et al. (2015).

As for the Eye of CSE categories, *Technical Solution* (91.7%) has been the one better covered by the organizations (which is consistent with the results considering the adoption degree) while the one less covered has been *Knowledge* (68.8%). When analyzing the adoption degree, the category with the lowest adoption was Quality. This suggests that although fewer practices related to *Knowledge* have been adopted, they have been adopted at higher levels than the ones related to Quality. These results are aligned with the top 3 practices less adopted by organizations, which showed that organizations have not been much worried about metrics, crucial to quality improvement.

When analyzing the distribution of the practices' adoption degree, we noticed that the adopted practices and the level at which they have been adopted have been more heterogeneous in the CI context (ranging from 15% to 70%, with predominance between 31% and 53%). We hypothesize that this may be a consequence of the different types of organizations involved in the study, which have different goals and business models that may influence the way some CI practices are performed. On the other hand, variation in RD practices adoption has been smaller (ranging from 18% to 44% with predominance between 29% and 41%). Although the behavior of the studied organizations is more homogeneous at this stage, practices at this stage have been adopted to lower degrees. Thus, in general, organizations have not yet reached continuous experimentation. Although the studied organizations are not committed to any software process maturity models, it is possible to notice that their behavior is in line with aspects of the general behavior described in such models.

Considering the whole scenario, we observed that the practices with the highest adoption degrees focus mainly on requirements, data storage, development, and team management, which are AO-related practices typically addressed in the initial levels of software process maturity models such as CMMI (CMMI, 2018) and MPS.BR (Conte et al., 2015). On the other hand, the practices with the lowest adoption degrees refer to experimentation, continuous process improvement and use of data, being mostly CD and RD practices. These practices are consistent with the ones performed in high levels of software process maturity models, which involve applying (advanced) statistical techniques to analyze collected data and support data-driven decision-making, performing simulations, and conducting pilots aiming at product and process improvement as well as innovation. When looking at the organizations individually (Figure 12), we note different scenarios. Only three organizations adopt (to some extent) all 71 practices. 11 organizations (93%) adopt more than 90% of the practices and four (14%) adopt less than half of the practices, with one of these organizations adopting less than one third of the practices. Moreover, some organizations have not adopted any practice related to some stages (e.g., organizations id 8, 16 and 25 in Figure 12 have not adopted R&D practices and organization 8 has also not adopted CD practices). The different scenarios found in this study reveal that CSE has been heterogeneously adopted in the studied organizations. On one hand, there are organizations that have advanced in CSE adoption. On the other, there are organizations at a very beginning stage.

Regarding *how different types of organization have adopted CSE (RQ2)*, in the three considered types, practices related to *Agile Organization* had the highest adoption degree and practices related to *R&D as an Experiment System* had the lowest one. This is consistent with the findings when analyzing the organizations as a whole. However, we can notice different results regarding *Continuous Integration* and *Continuous Deployment*. In Organizations with an IT Department, *Continuous Integration* had the second highest adoption degree, while in Software Houses and Startups it was *Continuous Deployment*. This is also noticed when analyz-

ing some practices individually. For example, *(CI.03) Builds occur frequently and automatically* and *(CI.06) Code is integrated constantly and automatically*, which are practices related to Continuous Integration, had a higher adoption degree in Organization with an IT Department than in the other organization types. On the other hand, *(CD.09) Marketing strategies are constantly evaluated and revised (when necessary) based on information from lead customers* and *(CD.10) Sales strategies are constantly evaluated and revised (when necessary) based on information from lead customers* had a higher adoption degree in Software Houses and Startups. We believe that these results are due to the fact that Software Houses and Startups deliver software for external clients, which can receive new versions of the product as new features are developed. On the other hand, Organizations with an IT Department develop software for their own organization and, thus, it may be more important to automatically integrate software than automatically deploy it. Moreover, since the organization develops software for itself, sales strategies probably do not apply.

By analyzing Figure 13, we note that Software House is the organization type with better results, being above the total average in all categories. This is probably related to the fact that Software Houses' business is focused on software development and, thus all investment is on improving software development skills. Different from Startups, which also focus on software development but aim at innovation, Software Houses usually have more established practices. The results showed a large advantage in practices related to Teams, which might be a result of higher efforts to keep competent and strategic people in the organization for a long time. The Startups scenario is close to the average, being a little better in Business and User/Customer, and not so good in Quality. This might be due to the focus on being well-positioned in their market to survive, which might cause organizations to neglect quality activities. The results suggest that more actions have been dedicated to Business, to design and implement products (Technical Solutions), and to constant interaction with the User/Client. Sometimes product quality suffers from this, intentionally or not. Organizations with an IT Department are also close to the average, but a little below, except for Quality. In this category, they are the best. This might be due to the relation with an internal client, which might result in more flexible deadlines, favoring quality aspects.

Concerning *Repeatable behaviors that have happened in CSE adoption (RQ3)*, we identified some practices that have (or not have) been adopted together. Among others, this was the case of *(CI.03) Builds occur frequently and automatically* and *(CI.06) Code is integrated constantly and automatically*; *(CD.09) Marketing strategies are constantly evaluated and revised (when necessary) based on information from lead customers*) and *(CD.10) Sales strategies are constantly evaluated and revised (when necessary) based on information from lead customers*; and *(RD.09) Data from the customer/consumer data repository is used in decision making by the software development area* and *(RD.10) Data from the customer/consumer data repository is used in decision-making by the business area*.

The found correlations reveal some expected behaviors. For example, if the organization performs frequent automatic builds, it is expected that the built code is automatically integrated. If the organization collects data from lead customers, it is expected that data is used to support sales and marketing strategies. Similarly, if the organization has a repository with user data, it is expected that it is used to support business and software development decisions. None of the correlations revealed unexpected behavior. Contrariwise, they reinforce knowledge about practices provided in the literature.

The results indicate that most of the organizations considered in the study have adopted CSE practices gradually, covering different stages and evolving according to the organizations' needs. This is indeed important because some practices may not be suitable for the organization business and, thus, should not be adopted by it. For example, practice CI.10 cited before only applies to organizations that have external agents working on software implementation. Otherwise, that practice is probably useless. Finally, although the StH model proposes a sequential and evolutionary path for CSE adoption, the results reveal that organizations have not followed that path systematically. The results are consistent with the arguments of Johanssen et al. (Johanssen et al., 2018) and Barcellos (Barcellos, 2020), who state that CSE adoption does not have a sequential nature and even if some CSE elements, such as continuous integration and delivery, require a step-wise introduction, CSE should be approached from multiple angles simultaneously. The CSE evolution in an organization it is more like a needs-oriented advance than a step-by-step path.

6 Going further on how Organizations have performed CSE

After analyzing the survey results, we invited everyone who responded to the survey to participate in an interview to provide further information about how CSE practices have been adopted in their organizations. Our goal was to get further information on how the organizations have performed CSE practices. Five participants agreed to participate in the interviews. We scheduled an online meeting with each of them.

The participants were the ones from organizations Org03, Org09, Org13, Org21, and Org24. Org09, established in 2017, is a Startup that offers a mobile financial control solution for individuals investing in the stock market. In 2023, its mobile solution boasted 1.2 million users. Org03, Org21, and Org24 are Organizations with an IT Department, and they operate within distinct sectors, respectively: cosmetic products (specifically, beauty products) commerce, car rental service, and steel industry. Org13 is a Software House that provides solutions encompassing a wide range of technologies, including artificial intelligence, mobile and web systems, and industrial automation. Table 8 summarizes information about the organizations' profile.

Based on data provided in the survey, Figure 17 provides an overview of the adoption degrees of CSE practices per StH stage in each organization that participated in the interviews. Figure 18, in turn, presents the adoption degrees of CSE practices per Eye of CSE category. These graphs depict

Table 8. Organizations' profile.

#	Type	Age (Y)	Region
Org03	Organization with an IT Department	45	South
Org09	Startup	5	Northeast
Org13	Software House	19	North
Org21	Organization with an IT Department	49	Southeast
Org24	Organization with an IT Department	38	Southeast

a panorama of the CSE scenario in each organization that participated in the interview.

Table 9 summarizes the participants' (here called P03, P09, P13, P21, and P24, i.e., we use the same id of the corresponding organization) profile. Two participants (P13 and P24) have a Master's degree, one (P21) is a Master's degree student, and the other two (P03 and P09) have a Bachelor's degree and did specialization courses. Two participants (P03 and P24) play lead roles in software development (Tech Leader), while one (P09) is a Director, and two (P13 and P21) are Developers. All declared to have knowledge of agile development, continuous integration, and continuous delivery. Three participants (P13, P21, and P24) declared a lack of knowledge of continuous experimentation. Concerning practical experience, all participants informed that have applied agile development and continuous integration in practice. P24 said that have not experienced continuous deployment and continuous experimentation. P13 also declared not have had practical experience with continuous experimentation.

6.1 Data Collection and Results

Data was collected in interviews from July to September 2023. The interviews were online and lasted about 60 minutes each. The interviewers were the first and the second authors. The interviews were recorded and transcribed.

We started the interview with each participant by explaining its purpose and recovering some information provided by the participant in the survey. The initial purpose was to recover the organizational scenario described by the participant in the survey, so that we could ask for details about how CSE practices have been performed and also verify if there had been changes in the organization since then. Considering that it would be unfeasible to ask about each of the 71 practices addressed in the survey, for each organization we identified the StH stages and Eye of CSE categories highlighted in Zeppelin's Analytic Report as the ones with the highest or the lowest adoption degrees and we focused our questions on them. However, we were not rigid about that. In the interviews, interviewees were told to feel free to talk as much as they wanted to. Thus, we let the interviewees talk as freely as possible, so they could provide the information they judged relevant without feeling limited by our questions.

We followed a semi-structured interview approach. For each organization, we defined some main questions considering the survey results and used these questions as the basis for the interview. In general, we pointed out the Eye of CSE categories and StH stages with the highest adoption degrees in the organization and asked the participant to talk about how its organization has performed practices related to them (e.g., "How does your organization perform practices related to «Eye of CSE category or StH stage»?). In some cases, we asked about specific practices (e.g., "How has «CSE practice» been performed in your organization?"). We also asked about practices related to the Eye of CSE categories and StH stages with the lowest adoption degrees. In this case, we focused on understanding the reasons for or consequences of not performing some practices or adopting them to a low degree. We used questions such as "Why does not your organization perform some practices related to «Eye of CSE category or StH stage»?", "Why your organization does not perform «CSE practice»?", "Are there any plans to adopt it/them in the future?". The script we followed in the interview is available in the study package (Santos Júnior et al., 2024). Next, we resume information gathered in the interviews.

Org03

Org03 was the one with the highest adoption degrees among the organizations studied in the survey (see Figure 17). Therefore, as there were many practices with high adoption degrees, when interviewing P03, instead of focusing on a specific StH stage or Eye of CSE category, we asked the participant to talk freely about how Org03 has performed practices related to each StH stage.

Concerning *Agile Organization*, Org03 performs agile practices (e.g., Scrum ceremonies, backlog, autonomous teams, software as value). P03 said that the most important achievement in this context in Org03 has been the alignment between development and business teams. P03 emphasized that the good alignment reached today is mainly due to the CSE practices implemented in the last years. P03 said that by keeping continuity throughout the whole process, development teams are not isolated islands and are now able to realize the impact of a new feature on technical and business indicators. P03 argued that sharing business results with development teams (e.g., by means of indicators) has significantly boosted developers' motivation and commitment. P03 cited the example of proactive behavior, which was not common in the past and has become frequent: *"one of the developers identified an issue in a feature of a product in the production-like environment and proactively resolved the problem because they knew the value of that feature for the organization"*. However, P03 pointed out that collecting data for the indicators may not be easy because of different understandings of the data or the indicators themselves. According to P03, it happened that the same indicators were presented with different values by development and business teams, which demanded clarifications and recalculation.

As for *Continuous Integration* and *Continuous Deployment*, P03 said that in Org03 a new code is integrated into the main code when test coverage is at least 80%. New features are delivered in a production-like environment on dates aligned between the development and business teams. The new feature is first delivered to an intermediate production-like environment, to which a limited group of users have access. If no issues arise in this environment, then the feature is deployed to the main real production environment, accessible to all users.

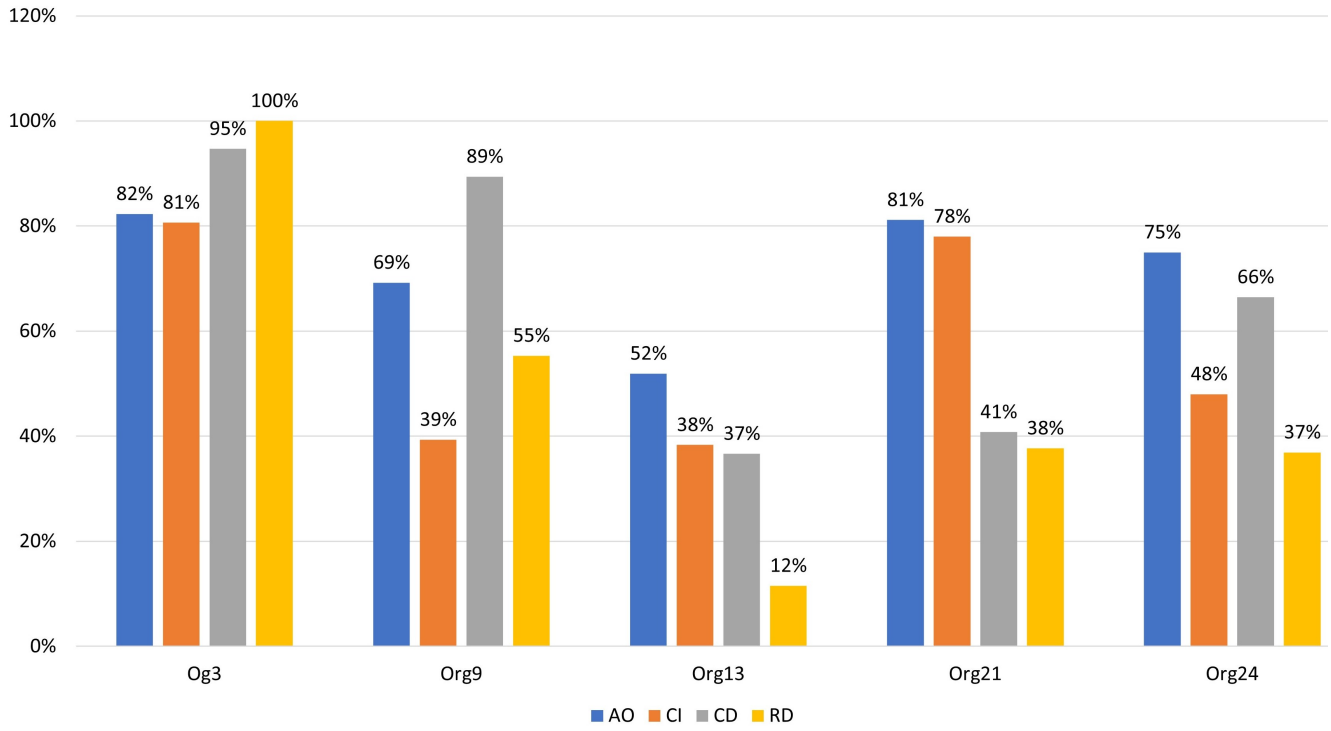

Figure 17. Adoption degree by StH stage per interviewed organization

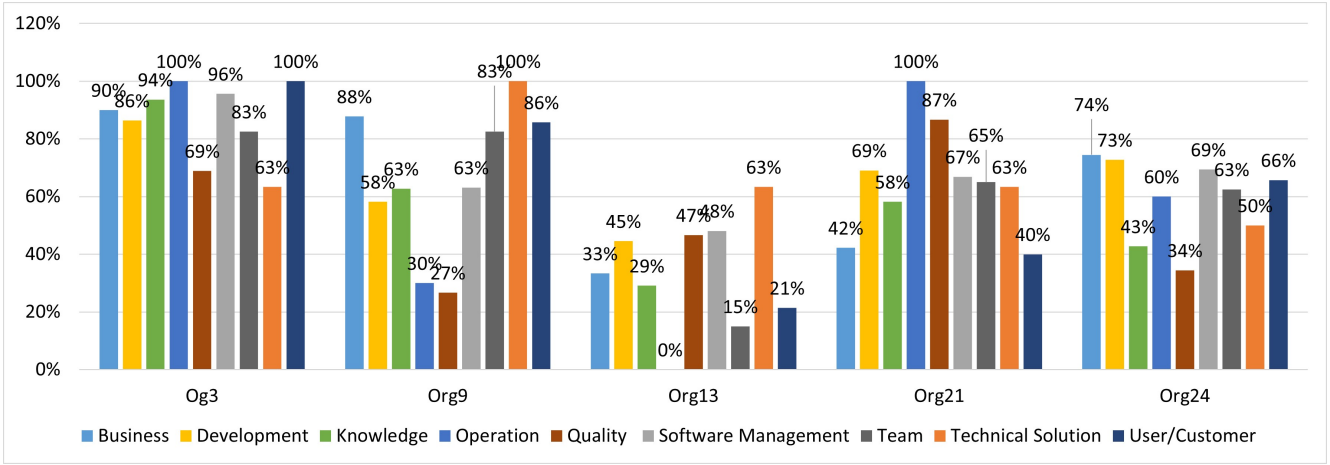

Figure 18. Adoption degree by Eye of CSE category per interviewed organization.

Table 9. Participant's profile.

Participant	Role	Degree	CSE	AO	CI	CD	RD
P03	Tech Leader	Graduate students with specialization courses	Knowledge	High	High	High	Medium
			Experience	High	Medium	High	Low
P09	Director	Graduate students with specialization courses	Knowledge	High	Medium	Medium	Medium
			Experience	High	Low	Low	Medium
P13	Developer	Master's Degree	Knowledge	Medium	Medium	Medium	None
			Experience	High	Medium	Medium	None
P21	Developer	Master's Degree Student	Knowledge	Medium	Low	Low	None
			Experience	High	High	High	Low
P24	Tech Leader	Master's Degree	Knowledge	Medium	Medium	Low	None
			Experience	High	Medium	None	None

R&D as an Experiment System has an adoption degree of 100%. That means that Org03 performs all practices of this stage in Zeppelin at institutionalized level (i.e., the practices are well-defined and used in all projects). P03 explained that Org03 has two teams that adopt experimentation practices (e.g., A/B Test, online feedback): the Discovery Team focuses on understanding customers needs, discovering new features or products, and getting new customers, while the UX Team is responsible for the product UX. According to P03, customer feedback and production-environment data are systematically used to improve processes and products.

When we asked P03 about CSE practices that they believe need to be improved in Org03, P03 pointed out that automated end-to-end tests have been challenging to implement. P03 also said that, when defining requirements, it may not be trivial to identify all the acceptance criteria necessary to determine if a user story is "done". These criteria are very important to ensure that the development area delivers features that earn value to the users. Therefore, properly defining them positively impacts the alignment between development and business. Finally, P03 mentioned that although there are many well-defined and standardized practices in Org03, there is no standard development process. Each team can choose the processes and technologies to be used. On one hand, this is positive because it contributes to team autonomy. On the other hand, it makes it more difficult to manage several teams, since they work in distinct ways. P03 pointed out that there has been an effort in Org03 to standardize the way of working aiming to improve the integration of several teams and facilitate CSE practices integration.

Org09

As can be seen in figures 17 and 18, Org09 has the highest adoption degree in practices related to *Continuous Deployment* and *Agile Organization*, from the StH stages perspective, and *Technical Solution* and *Business*, from the Eye of CSE categories perspective. Concerning *Agile Organization* and *Business*, P09 pointed out that Org09 performs processes and practices that enable agile development and the alignment between business and development teams in different parts of the development process. For example, in the sprint planning, Org09 seeks to ensure that only features that truly deliver value to the user can be added to the sprint backlog. According to P09, "*Defining development work items aligned with business goals provide the developers with the view of how a feature brings competitiveness to the organization. For example, if changing the color of a button increases user UX and engagement with the application, that feature is prioritized and added to the sprint backlog*". Moreover, development teams are set to product modules (or to internal products) instead of projects. In this way, each development team is responsible for developing, maintaining, and operating a product of a self-contained part of it, which helps bring development and business teams even closer.

The development teams are divided into front-end and back-end teams. According to P09, some time ago Org09 "*started granting more autonomy to team leaders"* and now "*teams are free to decide about the practices and architectures to be adopted*". Org09 has implemented the *Low Walls Culture*, which enables a developer to start a conversation with another developer, from a different team, to resolve an issue. According to P09, this "*resulted in the culture in which everyone is part of the solution"*. P09 also highlighted that teams in Org09 are motivated to "*self-reflection and discipline and continuously evolve skills*".

When talking about *Technical Solution*, P09 said that they adopt good programming practices, naming conventions (e.g., using *camelcase*[3]) and combine Domain-Driven Design (DDD) (Evans, 2004) with Behavior-Driven Development (BDD) (Nascimento et al., 2020). According to P09, "*the use of BDD and DDD has played a crucial role in facilitating maintenance and evolution*". Aiming to build suitable and innovative solutions, Org09 performs the *Weekly Forum* where senior members who compose a chapter get together to discuss technical issues, such as architecture standards, CSE practices to be performed, and supporting tools that can be used. The results are shared to be used by the teams.

Continuous Deployment is very important to Org09 because it has to frequently and automatically deploy new versions of the main product to the users. In addition, the product's new versions must be aligned with the Org09 business plan. In order to achieve that, a marketing and development team, constituted of people from development and marketing areas, is responsible for ensuring that the delivered product is aligned with the organization's goals and marketing plan.

[3] https://en.wikipedia.org/wiki/Camel_case

A new feature or product only is delivered to the user if there is alignment between development and business. For example, the delivery of a new feature can be aligned to business through a marketing campaign. Thus, the new feature only can be delivered to the user after the respective marketing campaign. According to P09, "*a lack of alignment between development and marketing would break development and business alignment and harm achieving the Org09 goals*".

CSE practices related to *Continuous Integration* (StH stage) and *Quality* (Eye of CSE category) have a lower adoption degree (see Figure 17 and Figure 18). When asked why Org09 has adopted CSE practices to a lower degree in these dimensions, P09 explained that regarding Continuous Integration, the initial version of the product was built using monolithic architecture and as the use demand increased, it was decided to change to microservices architeture, and "*it has been very challenging implement automated tests and continuous integration with microservices architecture*". Concerning *Quality*, P09 pointed out that they perform automated tests (to some degree), code coverage, and code review practices, but there are still some quality practices (e.g., use of collected data to improve processes) that need to be implemented and others (e.g., automated tests) that need improvements. They are often neglected to meet time-to-market.

Org13

From the five organizations participating in the interviews, Org13 is the one with the lowest adoption degrees. It is a software house that develops projects involving software and hardware and, thus, dealing with several technologies, such as artificial intelligence (AI), mobile and web applications, and automation, among others. The CSE practices focused by Org13 have been the ones at *Agile Organization* stage, followed by *Continuous Integration* and *Continuous Deployment*.

Org13 has a business agreement with a multinational organization and agreed to follow the software development process defined by the multinational organization, which is based on Scrum. Thus, the organization performs the well-known Scrum ceremonies, manages backlog, and has self-organized development teams. Although the process works well for the development team, it needs adaptations to incorporate the multidisciplinary team that involves people from development, AI, and automation. According to P13 "*while the Scrum-based process works well for the development team, the automation and AI teams face challenges"*. P13 pointed out that "*automation and AI teams work better with the waterfall process than the agile process*".

P13 informed that the organization adopts good practices of software development and that CI and CD practices have been incorporated into the projects (Org13 has its highest adoption degree in the Technical Solution category - see Figure 18). A production-like environment is used for the delivery of features before they are deployed in the production environment. P13 informed that there is a team (the deployment team) responsible for configuring the CI/CD supporting tool (e.g., creating scripts, jobs, and pipelines) and ensuring that deliveries and deployments are performed accordingly. Automated tests have been implemented recently. However, P13 highlighted that it has been difficult, particularly for the automation and AI teams.

Practices from the *R&D as an Experiment System* stage have been performed to a very low degree in Org13 (see Figure 17). P13 said that experiments have not been carried out because the team is not yet mature enough to deal with that (e.g., P13 said that at that moment the teams were focused on improving test practices). From the Eye of CSE categories perspective, the lowest adoption degrees of Org13 are in *Operation* and *Team*. P13 mentioned that to go forward in implementing CSE practices in Org13, it is necessary to deal with the particularities of the Org13 environment, which includes multidisciplinary projects and business agreements that impact software development. Moreover, they highlighted that Org13 is still learning how to do some practices and needs to get more mature to advance to new ones.

Org21

It has the highest adoption degrees in *Agile Organization* and *Continuous Integration* StH stages and *Operation* and *Quality* Eye of CSE categories. Concerning *Agile Organization*, P21 said that Org21 uses SAFe (Dean, 2016) to implement agile practices and culture (e.g., ceremonies, backlogs, autonomous teams, etc.) in different departments (Business, Development, and Operation). As a result, the development team is composed of business and development members. Each team is responsible for a product and is divided into two: one to develop the product and another to maintain it. P21 said "*It is common that a business person participates in sprint plannings and sprint reviews, which did not happen before we implemented SAFe*".

As for *Continuous Integration*, P21 pointed out that new code is integrated into the main code only if it passes the unit tests and has at least 80% approval on the test quality criteria. P21 said that "*unit test is a widespread practice in the organization, while the teams are still learning how to implement other types of tests (e.g., integration test)*". Org21 has a team responsible for developing the CI/CD pipelines for the projects. According to P21, "*this practice reduces problems related to CI and CD in the projects - for example, it avoids that a developer changes the CI configuration in the CI Server and affects integration in the project*".

P21 explained that concerning *Quality* aspects, Org21 has an expert team responsible for defining best practices, technologies, and process standards to be used by the development teams. However, Org21 does not have a role in the development team or a team in the organization dedicated to quality assurance. According to P21, this happens because Org21 has a culture in which all developers are responsible for ensuring quality. However, P21 believes that a person dedicated to quality assurance would be important to define and analyze quality criteria for processes and products of all development phases. As this role does not exist in Org21, P21 said that "*the development team implements the quality criteria based on what the team members think is important"*. P21 highlighted this issue as an aspect that should be improved in the future.

The Eye of CSE category with the highest adoption degree was *Operation*. P21 said that Org21 has a DevOps team that is responsible for creating and maintaining software artifacts and processes related to DevOps. When a new project

begins, it is possible to instantiate the environment containing all the established standards (e.g., architectural models, security definitions, CI/CD pipelines) and the development team can use it. According to P21, such practice allows the development team to focus on the product features (business rules, functions, user interface, etc.) while the DevOps team is concerned with CI/CD practices and standards.

The Eye of CSE category with the lowest adoption degree was *Business*. P21 said that Org21 needs to define clearer the responsibility of Business and Development teams. They commented that *"sometimes it is not clear who is responsible for a decision-making about a feature in a project"*. Moreover, P21 believes that it is necessary to reduce the distance between the development, business, and DevOps teams. Despite adopting CSE practices and aiming at continuity, the teams still do not interact with each other. For example, if the development team has a DevOps problem in a project, it is necessary to create a ticket for the DevOps team to fix the problem, as if it was a different department of Org21. P21 believes that *"it is still necessary that the teams be viewed as a single unit rather than being divided into separate departments (Development, Business, and Operations)"*.

Org24: It has the highest adoption degrees at *Agile Organization* and *Continuous Deployment* StH stages and *Business* Eye of CSE category. Concerning *Agile Organization*, Org24 performs agile practices based on Scrum. The development team may be located in two different organizational units, in different cities. The team is responsible for developing and maintaining applications to the Org24's departments, using a low code platform. There is a role responsible for quality assurance in the team. According to P24, "*introducing the quality analyst role in the development team was crucial to mitigate future maintenance issues and reduce maintenance demands*". *Continuous Deployment* is very important to Org24. P24 pointed out that *"The team is small and responsible for both the application development and maintenance. Thus, it is essential to ensure a smooth transition to production, without error, for the product-like environment. This way, the development team can prevent overburdening with maintenance issues"*. However, some projects do not have a quality analyst. According to P24, "*although the quality analyst brings benefits to the projects, the assignment of a person to this role is agreed with the client"*. As it means an increase in the project cost, P24 said that many times the client does not agree to pay for someone dedicated to quality assurance. P24 commented that it is common to hear the clients say "*Let me, the client, handle the tests"*.

Even performing some actions aiming at software quality, the *Quality* category is the one with the lowest adoption degree in Org24. P24 highlighted that it has been a challenge to implement some practices. P24 expressed a desire to implement automated testing: "*automated testing will be important to increase the quality during the development process, contributing to reduce future maintenance issues"*. According to P24, one of the difficulties in implementing testing practices is the Org21's culture. P24 commented that "*since project costs come out of the department's budget, clients are hesitant to allocate resources for testing"*.

Other CSE practices that P24 believes need improvement are the ones related to knowledge. P24 argued that "*Not having documentation makes it difficult to share knowledge with new team members and maintain projects in production-like environment"*. Lastly, P24 said that they have been working to implement *Pair Programming* and *Code Review* practices.

6.2 Discussion

The interview results provided us with additional information about how the interviewed organizations work. The five interviewed organizations have distinct CSE scenarios. Even so, we noticed that some practices remain common in all of them - particularly the ones related to agile. Similarly, some difficulties are faced by all the interviewed organizations (e.g., automated tests). Next, we consolidate and discuss data from all the interviewed organizations.

Agile Organization

We observed that all the organizations perform regular meetings (e.g., Scrum ceremonies), manage backlog, get business and development teams closer, and aim to deliver software as value. The need to align business and development was highlighted by several participants. In Org03, they use indicators to get development and business closer. In Org09, in turn, a business member participates in all the development phases to ensure that only features that truly deliver value to the user are delivered. The same happens in Org21, which implements SAFe practices. Additionally, in Org09, a market and development team is responsible for ensuring that the delivered product is aligned with the organization's goals and marketing plan. Org13 faces difficulties in implementing some agile practices due to the multidisciplinary nature of its projects. The teams involve people from development, automation, and artificial intelligence. Moreover, business and development alignment involves an agreement with a multinational organization, which represents some challenges to Org13.

In summary, all the participants acknowledge the importance of business and development alignment. In CSE context, this is empathized in the BizDev approach (Fitzgerald and Stol, 2017), which advocates that a continuous flow between business and development activities is necessary. This alignment is also necessary between development and operation, as defined in the DevOps approach (Leite et al., 2019). In fact, in CSE, a continuous flow between business, development, and operation (Fitzgerald and Stol, 2017) is crucial. However, as exemplified by P21, it is possible that organizations adopting CSE practices still have difficulties in achieving continuity in a wide spectrum. Moreover, the organizational structure also influences CSE practices. For example, Org21 has separated teams, which requires closer collaboration. Thus, beyond adopting CSE practices, it is necessary to change the organizational culture, structure, and understanding of the processes and teams involved in CSE.

Continuous Integration and Deployment

All the interviewed organizations perform some practices related to *Continuous Integration* and *Continuous Deployment*. In general, all of them consider that CI and CD processes help align development and business objectives and

offer a competitive advantage. Some practices are performed by all the interviewed organizations - e.g., all of them deliver new features or products in an intermediate production-like environment and, after that, deploy them in the actual production environment. However, there are some differences in how the practices are performed. For example, in Org03, a new code is integrated into the main code when test coverage is at least 80%, and new features are delivered in a production-like environment on dates defined by the development and business teams. In Org21, in turn, the new code is integrated into the main code if it passes the unit tests and has at least 80% approval on the test quality criteria. In Org09, deployment is aligned with business through a marketing plan. Thus, a new feature only can be deployed after the respective marketing action is performed. Moreover, some organizations have teams dedicated to CI/CD tasks. For example, in Org13, the deployment team is responsible for configuring the CI/CD supporting tool and ensuring that deliveries and deployments are performed accordingly. Similarly, in Org21, the DevOps team is responsible for creating and maintaining software artifacts and processes related to DevOps (e.g., architectural models, security issues, CI/CD pipelines). As we have observed in the survey results, automated tests have been a challenge. All the interviewed organizations report some difficulties. Automated end-to-end tests have been challenging for Org03. In Org09, they have faced difficulties in implementing automated tests in a microservices architecture. Automated tests have been implemented recently in Org13 and it has been difficult mainly due to the mix of technologies used in the projects (AI, automation, mobile, and web technologies). In Org21 automated unit tests are performed in all projects but they still have difficulties in implementing more robust automated tests, such as integration tests. Org24 does not perform automated tests. These results corroborate the survey results and indicate that, since CI involves automated tests and many organizations have not performed these tests properly, many organizations have not been able to truly perform CI. After all, a CI process uses automated testing activities to approve code integration (Shahin et al., 2017).

Some organizations pointed out concerns with quality aspects. Quality is crucial to ensure that the product meets the user's needs and is in conformance with the organization's quality standards. In other words, when performing CI and CD, it is necessary to ensure that the tests are efficient and effective as well as that the deployed product meets the user's needs and is aligned with the organization's goals. In Org21, there is a team responsible for defining best practices, technologies, and process standards. However, the development teams do not have a role dedicated to quality assurance - quality must be ensured by all developers. Differently, in the development teams of Org24, a person plays the role of quality analyst. This approach aligns with agile principles, which state that quality must be ensured by all developers. However, this role can be dismissed if the client is not willing to afford its cost, which, in the end, impacts the product quality. These results also reinforce the survey results, which suggest that practices related to quality have been performed to a low degree by the studied organizations and are often neglected due to time to market.

R&D as an Experiment System

As for *R&D as an Experiment System*, most of the interviewed organizations (Org13, Org21, and Org24) have the lowest adoption degree at this StH stage, which is consistent with the survey results. As we discussed in Section 5.2, a lower adoption degree at that stage is not surprising. However, Org09 stands out in these practices. It automatically and continuously collects user data and has teams that apply experiments and A/B tests to identify new features/products and make decisions related to the UX. During the interviews, when we asked the participants to talk about the RD practices the organization performs and the ones it does not perform, we noticed that in the survey the participants may have answered that the organization performs some RD practices to a certain level, when in fact it does not. For example, some RD practices contained in Zeppelin are statements that refer to the use of user/client/customer data (feedback) for decision-making and product/process improvement. Although some participants answered that the organization does that, during the interview, we noticed that in some cases data was not continuously and automatically collected, which is expected at that stage. It seems that even though Zeppelin provides a definition for each stage before presenting the respective practices, some participants may have not understood the statements properly. We believe that a more in-depth investigation of this aspect is therefore necessary.

7 Lessons Learned

In this section, we discuss some lessons we learned during this study. It is not our intention to provide an extensive list. Instead, we intend to share some of our perceptions. We adopt terms such as *should* and *may* instead of mandatory terms such as *must* because we believe that other studies are needed to corroborate what we have learned.

One size does not fit all. Different CSE can better suit different organizations. Thus, each organization should evaluate its needs and characteristics and implement CSE practices accordingly. In this sense, it is possible, for example, that certain organizations need a very robust and integrated suite of tools and detailed processes supporting collaborative software development and managing complexity. In contrast, there may be organizations that would be satisfied with a simpler CSE solution, focusing on some core practices that would be enough for the organization's purposes. Thus, it is necessary to select CSE practices according to the organization and project contexts.

Same CSE practice, different implementations. Even if organizations adopt the same CSE practice, they can implement it differently. Organizational and project characteristics (e.g., organizational structure, culture, business, customers, team knowledge and experience) can lead the same practice to be implemented in different ways. For example, daily meetings can be conducted differently if the organization has in-person, remote (with teams within the same country or in different countries), or hybrid development teams. Build and code integration strategies may depend on architectural and design choices. Thus, it is necessary to understand the different organizational aspects and how they relate to establish the

best (or more suitable) way to implement each CSE practice in the organization.

CSE practices should be gradually implemented. Implementing CSE at once is not trivial and may not be even possible. Therefore, CSE practices should be adopted gradually, focusing on leverage points that will make a difference in the organization. As some practices are consolidated, opportunities to implement new practices can be identified. Moreover, starting with one or two projects can be helpful. If these projects produce positive results, they can contribute to motivating the organization to extend the changes to other projects. In this way, besides increasing the number of adopted CSE practices, the organization can increase their level of adoption.

CSE practices should improve the organizational capability. Implementing CSE practices should not be a decision based on buzzwords. The implementation of CSE practices in an organization should increase the organizational capability, i.e., it should improve the organization's products, processes, and/or business, moving the organization from a situation to a better one. Metrics should be used to evaluate the effects of implementing CSE practices and support continuous improvement.

Implementing CSE practices involves bonuses and onus. On one side, implementing CSE practices contributes to improving the software development process and the produced software products. On the other side, it is necessary to spend effort to make the necessary changes in teams, workflows, tools, policies, and so on. People may be resistant to changes. Moreover, time and cost constraints can impact CSE implementation.

CSE involves culture change. CSE involves many technical practices, but also culture and structural changes. For example, a well-established end-to-end flow and jointing responsibility to deliver high-quality and valuable ready-to-use software are needed. Organizations interested in implementing CSE should be aware of and willing to such changes.

CSE goes beyond agile. CSE evolves agile development by preconizing an end-to-end flow that goes beyond development activities and considers a more comprehensive view of the SE processes, involving several organizational functions. This flow goes beyond the repetition of development activities organized in iterations. Therefore, to evolve from an agile to a continuous engineering approach, organizations should consider software development holistically.

8 Threats to Validity

The validity of a study denotes the trustworthiness of the results. Every study has threats that should be addressed as much as possible and considered together with the results. In this section, we discuss some threats considering the classification proposed in (Runeson et al., 2012).

Regarding *Construct Validity*, which is related to the constructs involved in the study, the main threat in the survey concerns the statements used to identify CSE practices in the questionnaire, which could be understood in different ways by different participants. To minimize this threat, we reviewed the sentences contained in *Zeppelin* (Santos Jr et al., 2021) aiming to make them clearer and we ran a test with two people to evaluate the questionnaire. This gave us the opportunity to minimize sources of misunderstanding. The set of CSE practices considered in the study is also a threat because it is not exhaustive and some CSE practices may not have been considered. We consider that this threat is minimized because we used *Zeppelin* (Santos Jr et al., 2021), which was defined based on works addressing CSE processes, activities, and practices ((Barcellos, 2020), (Fitzgerald and Stol, 2017), (Johanssen et al., 2018), and (Olsson et al., 2012)). Another threat refers to the weights assigned to the adoption levels in *Zeppelin*. This directly impacts the adoption degree calculation. If different weights are used, the quantitative results might be different. The questionnaire size is also a threat. It is long and participants may become confused and tired, influencing their responses.

Another important threat refers to the *Zeppelin* design. As explained before, *Zeppelin* organizes practices in stages based on the StH model and, thus, some practices often referred to as agile practices are located in CI or CD stages. Moreover, *Zeppelin* considers CI and CD as stages involving several practices that go beyond code integration and software deployment. These design decisions influence the results because the adoption degree is calculated by stage. Thus, if the practices were distributed differently, the adoption degrees could also be different. Hence, when interpreting the results, it is important to consider that, for example, when they reveal that the CI adoption degree is smaller than the CD adoption degree, that means that the organizations that participated in the study adopted the practices contained in the *Zeppelin's* CD stage at a higher degree than the practices contained in the *Zeppelin's* CI stage. Therefore, from Zeppelin's perspective, there has been more concern with the CD stage than the CI stage.

Another point that deserves attention is related to the knowledge and experience of the study participants. The participants declared their knowledge and experience levels and we did not establish any way to verify them. Thus, the lack of experience or knowledge may have influenced the answers. To mitigate this threat, we performed interviews with some participants. However, only five participants were interviewed.

Concerning the interviews, the participants may have misunderstood some questions or felt constrained to provide answers only to the questions presented by the interviewers. To address these threats, the interviewees were told to be free to talk as they wanted and to ask for clarifications if they did not understand any questions. Another threat refers to the fact that interviewing is time-consuming. It can make the interviewee tired, influencing their answers. To avoid this, we selected some CSE stages and categories to be approached in the interviews, which were limited to 60 minutes. Another threat to both, survey and interviews, relies on the fact that the provided responses may have been influenced by personal perception, which is directly influenced by the person's knowledge and experience with the subjects addressed in the study. Therefore, if different people represented the organizations considered in the studies, the responses could have varied. Therefore, the bias of the respondents should be considered as a threat.

Regarding *Internal Validity*, which is concerned with the relationship between results and the applied treatment, the time available for the participants to answer the questionnaire may have influenced the survey results. When we made the questionnaire available, the deadline for answering it was 15 days. However, we noticed that more time was needed to reach organizations from all Brazilian geographic regions. Thus, we extended the deadline so that each participant could have at least one week to answer the questionnaire. In the interviews, as the participation is not anonymous (interviewer and interviewee interact with each other in a meeting), the participants might be not willing to give honest answers because of afraid of being personally evaluated. We addressed this threat by clarifying that nobody would be judged and the answers would be kept anonymous. Another threat regards the fact that one interviewee may be more willing to answer than another. Thus, the answers provided by them will have different depths and comprehensiveness, which influence the results. Additionally, the interviewee can hide information about the organization due to confidentiality issues or being afraid of reprimand for revealing the organization's weaknesses or differential practices.

As for *External Validity*, which is concerned with to what extent it is possible to generalize the results, the main threat in the survey is the small number of organizations and most of them are from the same Brazilian macro-region. Although the sample reflects in some way the distribution of software organizations in Brazil (the Southeast region concentrates the higher number of organizations), ideally, the sample should be larger and the geographic distribution of the organizations more diverse. As for the interviews, the sample is even smaller, which constraints the results. Moreover, in the interviews, we considered only some of the CSE practices addressed by *Zeppelin*. Thus, other organizations could provide different results.

Finally, with respect to *Reliability Validity*, which refers to what extent data and analysis depend on a specific researcher, the main threat is that data analysis was performed by the authors. To minimize this threat, in the survey, analysis was carried out by two of the authors and reviewed by the other two. Data from the interviews were analyzed by two authors and reviewed by a third one. Discussions were performed until consensus.

In summary, considering all the mentioned threats, we can only present some insights from the results and generalization is limited. Thus, obtained results cannot be considered conclusive, but preliminary evidence of how CSE has been adopted in Brazilian software organizations.

9 Final Considerations

Currently, terms such as DevOps, Continuous Integration, and Continuous Deployment have been part of daily activities in several organizations. The 'continuous' phenomenon clearly indicates a common trend, namely the increasing need to establish an end-to-end flow between customer demand and the fast delivery of a product or service. The big picture by which this might be achieved goes beyond agile practices and surfaces a more holistic set of continuous activities (Fitzgerald and Stol, 2017). CSE is about performing various practices that integrate business and software development and experimenting with the customer (Karvonen et al., 2016).

This paper presented a study that aimed to provide a preliminary panoramic picture of how Brazilian organizations have adopted CSE practices. First, we conducted a survey by using *Zeppelin* (Santos Jr et al., 2021), a diagnostic instrument that supports identifying CSE practices adopted in an organization, their adoption degree, and their relation with StH stages in the CSE evolutionary path. The survey was applied to 28 Brazilian software organizations. In summary, the results showed that organizations have better covered *Agile Organization* and *Continuous Deployment* practices than *Continuous Integration* and *R&D as an Experiment System*. However, this scenario changes a bit in different types of organizations. Startups and Software Houses have focused more on *Continuous Deployment* than *Continuous Integration*, while the opposite was perceived in Organizations with an IT Department. Organizations have also adopted many practices related to Technical Solution, while Quality and Knowledge-related practices have been adopted in lower degrees. The adoption of CSE has been heterogeneous, but some repeatable behaviors revealed practices that have been adopted (or not adopted) together. After the survey, we interviewed five organizations to understand how they have performed CSE. The results showed that although the five interviewed organizations have distinct CSE scenarios and perform different practices, some practices are common in all of them - particularly agile and some CI and CD practices. Moreover, some difficulties are faced by all of them. In this matter, automated tests stand out.

The main conclusion we can reach from the results is that organizations have adopted CSE practices in different ways. Some practices have been adopted in most organizations, while others have been performed in only some of them. Moreover, some organizations have adopted CSE in a comprehensive way and, in contrast, others have focused on a subset of CSE practices. We believe that organizations have tended to follow a path suitable for them, by adopting practices gradually, covering different stages and categories, and evolving according to their needs. We also observed that some key CSE practices, such as automated tests, have been neglected by many organizations. Therefore, several organizations have not properly performed some CSE practices. In this context, there are still many challenges to overcome (the results pointed out some of them, such as lack of continuous experimentation and quality-related issues).

As future work, we intend to perform new studies involving other organizations to grow the sample size and representativeness, and improve the picture of CSE obtained from the results of this study. Moreover, we plan to go deeper into specific CSE practices (e.g., the ones related to CI and CD) to better understand how they have been adopted or why they have not been adopted (e.g., continuous experimentation). Additionally, we intend to use techniques such as Grounded Theory to codify qualitative collected data and provide finer qualitative results. As for the *Zeppelin*, we aim to produce an improved version, reviewing the practices (e.g., improving existing ones and adding others) and including detailed

explanations for the practices to better assist organizations in answering the questionnaire and identifying their position in the CSE path.

Acknowledgements

The authors thank all the survey and interview participants and also all the people who helped by suggesting organizations to participate in the study or spreading the study call for participation. This research is supported by the Espírito Santo Research and Innovation Support Foundation - FAPES (Process 2023-5L1FC and T.O. 1022/2022).

References

Barcellos, M. P. (2020). Towards a framework for continuous software engineering. In *Proceedings of the 34th Brazilian Symposium on Software Engineering*, SBES '20, page 626–631, New York, NY, USA. Association for Computing Machinery, https://doi.org/10.1145/3422392.3422469.

Beck, K. (2000). Extreme programming explained: embrace change. Addison-Wesley Professional.

Bosch, J. (2014). Continuous software engineering: An introduction. In Bosch, J., editor, *Continuous Software Engineering*, pages 3–13, Cham. Springer International Publishing, https://doi.org/10.1007/978-3-319-11283-1_1.

CMMI, I. (2018). Capability maturity model integration 2.0, https://cmmiinstitute.com. Accessed: 2023-Oct-29.

Conte, T. U., Oliveira, N. H. F., Prikladnicki, R., Rocha, A. R. C., Santos, G., Travassos, G. H., and Weber, K. C. (2015). Towards successful software process improvement initiatives: Experiences from the battlefield. In *American Conference on Information Systems*.

Dean, L. (2016). Safe® 4.0 reference guide: Scaled agile framework® for lean software and systems engineering. Addison-Wesley Professional.

Debois, P., Humble, J., Molesky, J., Shamow, E., Fitzpatrick, L., Dillon, M., Phifer, B., and DeGrandis, D. (2011). Devops: A software revolution in the making. volume 24, pages 3–39. Journal of Information Technology Management.

Easterbrook, S., Singer, J., Storey, M.-A., and Damian, D. (2008). Selecting empirical methods for software engineering research. In Shull, F., Singer, J., and Sjøberg, D. I. K., editors, *Guide to Advanced Empirical Software Engineering*, pages 285–311, London. Springer London, https://doi.org/10.1007/978-1-84800-044-5_11.

Evans, E. (2004). Domain-driven design: Tackling complexity in the heart of software. Addison-Wesley.

Fitzgerald, B. and Stol, K.-J. (2017). Continuous software engineering: A roadmap and agenda. volume 123, pages 176–189. Journal of Systems and Software, https://doi.org/10.1016/j.jss.2015.06.063.

Johanssen, J. O., Kleebaum, A., Paech, B., and Bruegge, B. (2018). Practitioners' eye on continuous software engineering: An interview study. In *Proceedings of the 2018 International Conference on Software and System Process*, ICSSP '18, page 41–50, New York, NY, USA. Association for Computing Machinery, https://doi.org/10.1145/3202710.3203150.

Karvonen, T., Lwakatare, L. E., Sauvola, T., Bosch, J., Olsson, H. H., Kuvaja, P., and Oivo, M. (2015). Hitting the target: Practices for moving toward innovation experiment systems. In Fernandes, J. M., Machado, R. J., and Wnuk, K., editors, *Software Business*, pages 117–131, Cham. Springer International Publishing, https://doi.org/10.1007/978-3-319-19593-3_10.

Karvonen, T., Suomalainen, T., Juntunen, M., Sauvola, T., Kuvaja, P., and Oivo, M. (2016). The crusoe framework: A holistic approach to analysing prerequisites for continuous software engineering. In *Product-Focused Software Process Improvement*, pages 643–661, Cham. Springer International Publishing, https://doi.org/10.1007/978-3-319-49094-6_52.

Leite, L., Lago, N., Melo, C., Kon, F., and Meirelles, P. (2023). A theory of organizational structures for development and infrastructure professionals. *IEEE Transactions on Software Engineering*, 49(4):1898–1911.

Leite, L., Rocha, C., Kon, F., Milojicic, D., and Meirelles, P. (2019). A survey of devops concepts and challenges. volume 52, page 35, New York, NY, USA. Association for Computing Machinery, https://doi.org/10.1145/3359981.

Myers, L. and Sirois, M. J. (2004). Spearman correlation coefficients, differences between. volume 12. Wiley Online Library, https://doi.org/10.1002/0471667196.ess5050.pub2.

Nascimento, N., Santos, A. R., Sales, A., and Chanin, R. (2020). Behavior-driven development: A case study on its impacts on agile development teams. In *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, ICSEW'20, page 109–116, New York, NY, USA. Association for Computing Machinery, https://doi.org/10.1145/3387940.3391480.

Olsson, H. H., Alahyari, H., and Bosch, J. (2012). Climbing the "stairway to heaven" –a mulitiple-case study exploring barriers in the transition from agile development towards continuous deployment of software. In *38th Euromicro Conf.on Software Engineering and Advanced Applications, https://doi.org/10.1109/SEAA.2012.54*, pages 392–399.

Runeson, P., Host, M., Rainer, A., and Regnell, B. (2012). Case study research in software engineering: Guidelines and examples. John Wiley & Sons.

Santos Jr, P. S., Barcellos, M. P., and Calhau, R. F. (2020). Am i going to heaven? first step climbing the stairway to heaven model results from a case study in industry. In *Proceedings of the 34th Brazilian Symposium on Software Engineering*, SBES '20, page 309–318, New York, NY, USA. Association for Computing Machinery, https://doi.org/10.1145/3422392.3422406.

Santos Jr, P. S., Barcellos, M. P., and Ruy, F. B. (2021). Tell me: Am i going to heaven? a diagnosis instrument of continuous software engineering practices adoption. In *Evaluation and Assessment in Software Engineering*, EASE 2021, page 30–39, New York, NY, USA. Association for Computing Machinery,

https://doi.org/10.1145/3463274.3463324.

Santos Jr, P. S., Barcellos, M. P., Ruy, F. B., and Omêna, M. S. (2022a). Flying over brazilian organizations with zeppelin: A preliminary panoramic picture of continuous software engineering. In *Proceedings of the XXXVI Brazilian Symposium on Software Engineering*, SBES '22, page 279–288, New York, NY, USA. Association for Computing Machinery, https://doi.org/10.1145/3555228.3555234.

Santos Jr, P. S., Barcellos, M. P., Ruy, F. B., and Omêna, M. S. (2022b). Supplementary material of the study "flying over brazilian organizations with zeppelin: A preliminary panoramic picture of continuous software engineering", https://doi.org/10.5281/zenodo.6857220.

Santos Júnior, P. S., Barcellos, M. P., Ruy, F. B., and Omena, M. S. (2024). Supplementary material of the study "A Preliminary Panoramic View of Continuous Software Engineering Adoption in Brazilian Organizations".

Shahin, M., Babar, M. A., and Zhu, L. (2017). Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices. In CoRR, http://arxiv.org/abs/1703.07019.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). Experimentation in software engineering. Springer Science & Business Media, https://link.springer.com/book/10.1007/978-3-642-29044-2.