Flying over Brazilian Organizations with Zeppelin

A Preliminary Panoramic Picture of Continuous Software Engineering

Paulo Sérgio dos Santos Júnior
LEDS Research, Department of Informatics, Federal Institute of Education, Science and Technology of Espírito Santo
Serra, ES, Brazil
paulo.junior@ifes.edu.br

Monalessa P. Barcellos
Ontology and Conceptual Modeling Research Group (NEMO), Computer Science Department, Federal University of Espírito Santo
Vitória, ES, Brazil
monalessa@inf.ufes.br

Fabiano B. Ruy
LEDS Research, Department of Informatics, Federal Institute of Education, Science and Technology of Espírito Santo
Serra, ES, Brazil
fabianoruy@ifes.edu.br

Moisés S. Omêna
LEDS Research, Department of Informatics, Federal Institute of Education, Science and Technology of Espírito Santo
Serra, ES, Brazil
omenam@ifes.edu.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. *Results*: The results indicate that organizations have better addressed agile and continuous deployment practices than the ones related to continuous integration and continuous experimentation, but this scenario changes a bit depending on the organization type. They also show that CSE adoption has been heterogeneous, but there are patterns in the adoption of some practices. *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 needs.

SBES 2022, October 5–7, 2022, Uberlândia,MG - Brazil

ACM ISBN 978-1-4503-9735-3/22/10…$15.00
https://doi.org/10.1145/3555228.3555234

CCS CONCEPTS

• **Software and its engineering → Software creation and management**.

KEYWORDS

Continuous Software Engineering , Stairway to Heaven, Survey

ACM Reference Format:
Paulo Sérgio dos Santos Júnior, Monalessa P. Barcellos, Fabiano B. Ruy, and Moisés S. Omêna. 2022. Flying over Brazilian Organizations with Zeppelin: A Preliminary Panoramic Picture of Continuous Software Engineering. In *XXXVI Brazilian Symposium on Software Engineering (SBES 2022), October 5–7, 2022, Virtual Event, Brazil.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3555228.3555234

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 [6] [1]. Hence, organizations should evolve to continuous and data-driven development in a continuous software engineering approach [1] [3].

Continuous Software Engineering (CSE) aims to establish a continuous flow between software-related activities, taking the entire software life cycle into account. It 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 [6]. In this context, some emerging initiatives have been proposed, such as Continuous Integration [2], which seeks to eliminate discontinuities between development and delivery; DevOps [4], which recognizes that the integration between software development and

software operation must be continuous; and BizDev [6], which 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., [12] [7] [6][1]). Although these works provide useful knowledge, they are not enough for organizations to identify which practices they should perform to implement a certain process or stage and how the organization can evolve in the CSE journey. Considering that, Santos Jr. et al. [16] proposed a diagnostic instrument called *Zeppelin*[1] to aid organizations in identifying their position in relation to 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 [12] as reference model and also considers Continuous* activities defined in [6], CSE elements and categories provided in the Eye of CSE [7] and CSE processes constituting the CSE framework proposed in [1].

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. In summary, the results revealed that organizations have better addressed agile 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.

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 advance of CSE in Brazilian organizations (e.g., to 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 organized as follows: Section 2 presents the theoretical background for the paper by addressing CSE and introducing Zeppelin, and also discusses some related work; Section 3 presents the study planning; Section 4 regards the study execution and results; Section 5 discusses the results; Section 6 addresses threats to validity; and, finally, Section 7 presents some final considerations and future work.

2 BACKGROUND

2.1 Continuous Software Engineering

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 [6]. According to Johanssen et al.[7], 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. [12] defined the *StH Model*, which describes a five-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 *R&D as Innovation 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.

From interviews performed with CSE practitioners, Johanssen et al. [7] 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.

Fitzgerald and Stol [6] 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

[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 [16].

life cycle. Continuous planning, continuous security, continuous use, continuous trust, and continuous experimentation are some of the considered Continuous* activities.

Finally, Barcellos [1] proposes a framework containing 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. Processes suggested in [6], elements from the Eye of CSE [7] and StH stages were considered to define *FCSE* [1]. Differently from StH, the framework considers that processes can be performed simultaneously and gradually.

Some studies have investigated the use of CSE practices in organizations. For example, Leite et al. [10] 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. The two studies we found most similar to ours are [9] and [14].

Karvonen et al. [9] 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, [9] also used StH as a reference model and evaluated at which level CSE practices were adopted. However, different from [9], which involves case studies in five Finnish organizations, our study consists of a survey with 28 organizations and aim to provide a preliminary panorama of CSE adoption in Brazilian organizations. Moreover, our study was perfomed by using Zepellin [16], which is based on StH and other works ([6] [7] [1]) not considered in [9].

Santos Jr. et al.[14] also evaluated the use of CSE in organizations by applying Zeppelin. However, different from our current study, the purpose was to evaluate five organizations individually (not to provide a panorama about CSE adoption) and show Zeppelin feasibility and usefulness.

2.2 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 [14]. 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. [16] proposed *Zeppelin*. It is a diagnostic instrument made up of two components: *Diagnostic Questionnaire*, which identifies the CSE practices an organization performs and their adoption degree; and *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 Innovation System (RD)* (13 practices). Besides being grouped by the StH stages, the practices are also categorized considering the Eye of CSE [7] dimensions (categories and elements).The practices were identified based on the literature and (mainly on [1], [6], [7] and [12]) and on practical experiences.

The questionnaire is used to evaluate which practices have been adopted in the organization and understand how comprehensive their adoption has been. When applying Zeppelin, for each statement, the user must indicate the level at which the referred CSE practice is adopted in the organization. The adoption levels were defined based on [9] and are used to capture the comprehensiveness of each practice in the organization and help monitor its evolution. *Not Adopted* level is used to identify practices that the organization has never used. *Abandoned* level refers to practices that were discontinued. *Project/Product* level is used to identify practices not formalized in the organization and used only in a particular project or product. *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 *Institutionalized* when it is formally defined and used in all projects.

The *adoption degree* at each stage (AD) is represented as a percentage and it 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, ADstage = (weightALpractice1 + ... + weightALpracticen) / n) * 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).

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 considering the StH stages [12]) and the dimensions of the Eye of CSE [7].Table 1 presents some statements of the Diagnostic Questionnaire and their related stage. Further information about Zeppelin can be found in [16].

3 STUDY PLANNING

The study consisted of a survey whose **goal** was to investigate the adoption of CSE practices in Brazilian organizations to get a preliminary panoramic view of CSE adoption in Brazil. A survey aims at identifying the characteristics of a broad population by generalizing on the data collected from a representative sample of individuals [5]. Surveys are conducted to produce a snapshot of the situation to capture the current status [18]. 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 interviews or case studies, for example, it would be unfeasible.

Aligned with the study goal, we defined the following main **research question**: *(RQ1) How has CSE been adopted in Brazilian*

Table 1: Examples of CSE practices contained in the Diagnostic Questionnaire.

StH Stage	Statement
Agile Organization	AO.04 For delivering value to the customer, requirements are defined and prioritized according to customer needs, are periodically reviewed, and changes are absorbed into iterations of the development process.
	AO.05 The scope of the project is defined gradually, using the Product Backlog (or equivalent artifact).
	AO.08 The development process is performed iteratively, in short cycles (e.g., 2 weeks), in which selected requirements recorded in a Sprint Backlog (or equivalent artifact) are developed.
Continuous Integration	CI.03 Builds occur frequently and automatically.
	CI.06 Code is integrated constantly and automatically.
	CI.13 The organization adopts practices that allow external organizations to act in the development of the project.
Continuous Deployment	CD.09 Marketing strategies are constantly evaluated and revised (when necessary) based on information from lead customers.
	CD.10 Sales strategies are constantly evaluated and revised (when necessary) based on information from lead customers.
R&D as Innovation System	RD.10 Data from the customer/consumer data repository is used in decision-making by the software development area.
	RD.11 Data from the customer/consumer data repository is used in decision-making by the business area.

software organizations, considering the StH stages and the Eye of CSE categories? This question aims at giving a general view of CSE adoption in Brazilian organizations. StH stages and Eye of CSE categories are used to provide the general view considering different perspectives. We considered these perspectives because they are the ones addressed by Zeppelin. From this question, we defined other two 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?* and *(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. We needed to adapt the version available in [16] to make it feasible to apply it in the study. The Zeppelin version proposed in [16] 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 [16] 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.

The **procedure** followed in the study consisted of four 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 is 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 final step, we sent the analytic report to each study participant.The form used in the study is in the study package available in [15].

The **participants** of the study were people with knowledge of and experience in CSE and that work in Brazilian organizations that perform CSE practices. It was allowed the participation of only one person representing each organization.

4 STUDY EXECUTION AND RESULTS

In this section, we present information about the study execution and show some of the collected data. Discussion about data will be made in Section 5.

The Zeppelin questionnaire was made available in March 2022 and data was collected until late April 2022. The study involved

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	7.1%
Between 10 and 49 employees	5	17.9%
Between 50 and 99 employees	3	10.7%
More than 99 employees	18	64.3%
Region	**Quantity**	**%**
Midwest	2	6%
North	3	11%
Northeast	3	11%
South	3	11%
Southeast	17	61%

28 participants from different Brazilian organizations. In fact, two participants were from the same company but worked in different and separated units, thus we considered them as different organizations. 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 has a repeatable and scalable business model), and 25% are *Software Houses* (i.e., organizations that develop software for other organizations). Considering size, 7.1% of the organizations have between 1 and 9 employees, 17.9% have between 10 and 49 employees, 10.9% have between 50 and 99 employees, and 64.3% 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. Table 2 summarizes organizations type, size and region.

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. The answers were informed 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. In the following, we present a data synthesis for each research question.

Table 3: Participant's Knowledge and Experience.

	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: How has CSE been adopted in Brazilian software organizations, considering the StH stages and the Eye of CSE categories?

For answering 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 Section 2),to provide a view of CSE adoption based on the level at which the practices are adopted (Produt/Project, Process, Institutionalized, Not Adopted, 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 the organizations.

Figure 1 shows the adoption degree related to each StH stage. The adoption degree was calculated using the procedure established in Zeppelin (see Section 2.2). As it 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%).

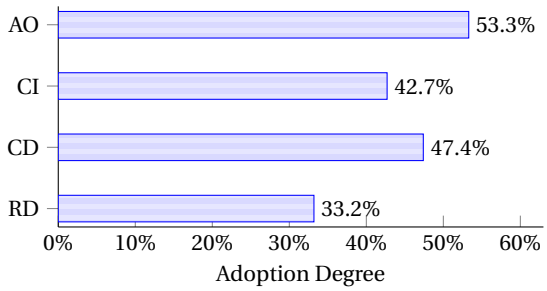

Figure 1: Adoption Degree per StH stage.

Figure 2 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%).

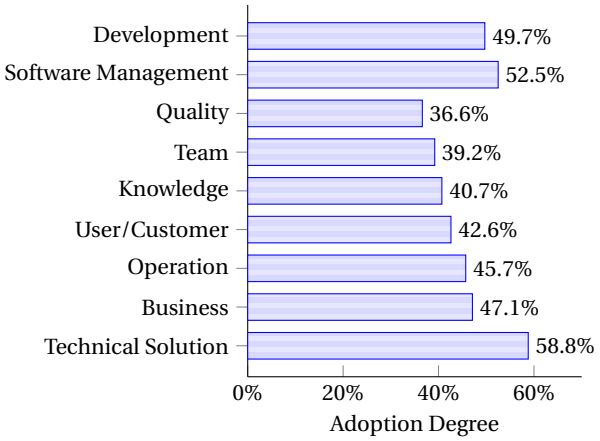

Figure 2: Adoption Degree per Eye of CSE category.

Aiming to represent the level at which CSE practices have been adopted, Figure 3 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 4 provides an overview of the practices adoption per stage, regardless the level at which the practice is adopted. Thus, data presented in Figure 4 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 can be found in [15].

To provide a view of practices adoption by each organization, Figure 5 presents the number of practices adopted by each participant Organization at each StH stage.

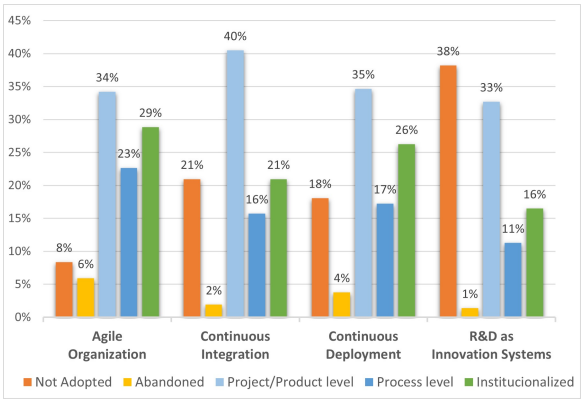

Figure 3: Practices per Adoption Level.

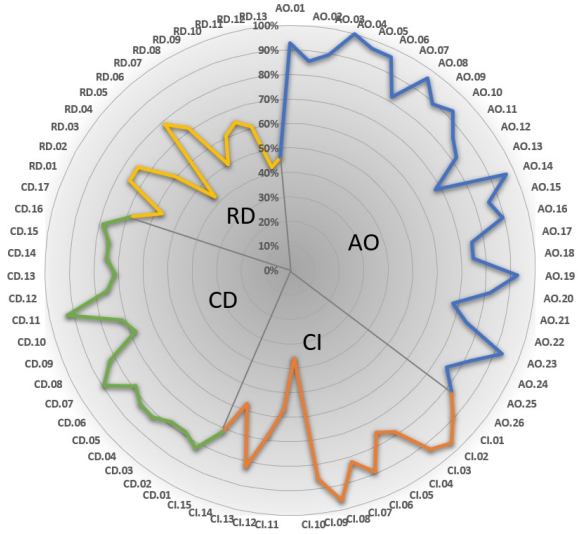

Figure 4: CSE Practices Adoption.

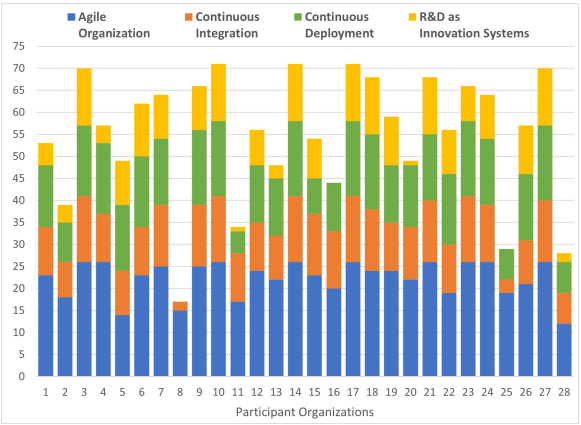

Figure 5: CSE Practices Adoption per Organization.

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 and size. Table 4 presents the adoption degree at each StH stage per organization type. As it 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 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

Table 4: 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%**

Table 5: 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%**

in Quality. Table 5 summarizes the adoption degrees by organization type and at each category. Figure 6 illustrates data from Table 5 providing a general view of the adoption degrees per organization type and compared to the total adoption degree (i.e., considering all organization types).

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 coefficient [11] 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 present three of them. Others can be found in [15].

Figure 7 shows the correlation between two practices related to CI: *(CI.03) Builds occur frequently and automatically* and *(CI.06) Code is integrated constantly 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

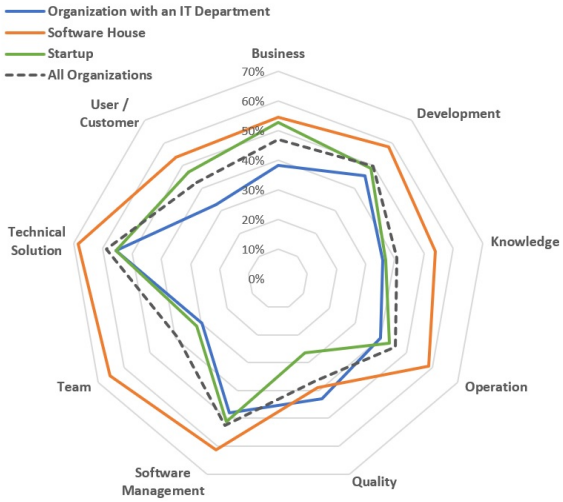

Figure 6: Adoption Degree per Organization Type and StH Stage

adoption degree of the two considered CSE practices. For the CSE practices represented in Figure 7, 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.

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*. In this case, spearman=0.944 and R^2=0.905.

As for practices related to Research & Development as Innovation 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.

5 STUDY ANALYSIS AND INTERPRETATION

Aiming at answering the research questions defined in the study, in this section we analyze and discuss data presented in Section 4. 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 Innovation 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

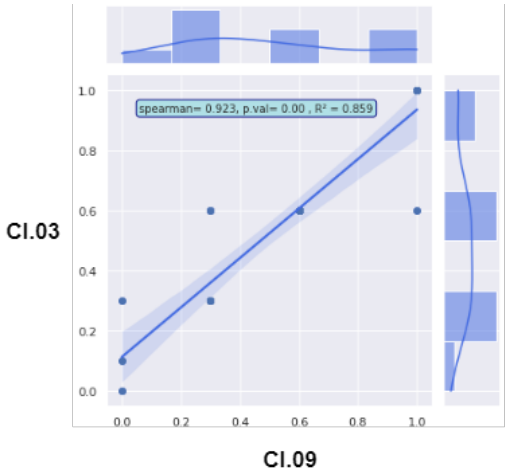

Figure 7: Correlation between CI.03 and CI.06.

make improvements at software development and business levels [12]. 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, it is crucial that the code is tested in a proper test environment [17].

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 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 test 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 at providing a complementary view, we also analyzed data to identify practices that have been adopted by the organizations, regardless 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) In order to deliver value to the customer, requirements are defined and prioritized according to customer needs, are periodically reviewed, and changes are absorbed into iterations of the development process)*, and one adopted by only ten organizations (*(CI.10)There are practices that allow organizations or people outside the project to act in the implementation of the product (i.e., produce and integrate code into the product being developed))*. We noticed that the percentage of adopted practices is not homogeneous.

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 Innovation 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. [9].

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 less practices related to Knowledge have been adopted, they have been adopted at higher levels than the ones related to Quality.

When looking at the organizations individually (Figure 5), 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 5 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 Innovation 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 analyzing 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 the 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 6, 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 aiming 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 in order to survive, which might take organizations to neglect quality activities. The results suggest that more actions have been dedicated to Business, to design and implement products (Technical Solution) and to constant interaction with the User/Client. Sometimes product quality suffers from this, intentionally or not. Organizations whit an IT Department are also close to the average, but being a little below, except for Quality. In this category, they are the best. This may be due to the relation with an internal client, which may 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. [7] and Barcellos [1], who state that CSE adoption has not 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 a organization it is more like a needs-oriented advance than a step-by-step path.

6 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 [13].

Regarding *Construct Validity*, which is related to the constructs involved in the study, the main threat 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 [16] aiming to make them clearer and we ran a test with two people to evaluate the questionnaire. This gave us an 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 have not been considered. We consider that this threat is minimized because we used Zeppelin [16], which was defined based on works addressing CSE processes and practices ([1], [6], [7], and [12]). 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 may be a little different.

Concerning *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 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 for answering the questionnaire.

As for *External Validity*, which is concerned with to what extent it is possible to generalize the results, the main threats in this study 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 larger number

of organizations), ideally, the sample should be larger and the geographic distribution of the organizations more diverse.

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, analysis was carried out by two of the authors and reviewed by the other two. Discussions were performed until consensus.

In summary, considering all 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.

7 FINAL CONSIDERATIONS

Currently, terms such as DevOps, Continuous Integration and Continuous Deployment have been part of daily activities in several organizations. The 'continuous' phenomena 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 [6]. CSE is about performing various practices that integrate business and software development and experimenting with the customer [8].

This paper presented a study that aimed to provide a preliminary panoramic picture of how Brazilian organizations have adopted CSE practices. The study was conducted by using Zeppelin [16], 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 best covered Agile and Continuous Deployment practices than Continuous Integration and R&D as Innovation 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.

We believe that the main conclusion we can reach from the study results is that there has not been a "one and right path" to adopting CSE. 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. In this context, there are still many challenges to overcome (the study results pointed out some of them, such as lack of continuous experimentation and quality issues).

As future work, we plan to go deeper into this study and carry out specific analysis to better understand how some CSE practices have been adopted and the reasons why some practices have not been adopted (e.g., Are organizations facing difficulties to implement them? Are some practices less suitable for some organizations?). We also intend to perform new studies involving other organizations to grow the sample size and representativeness, and improve the picture of CSE in Brazil obtained from the results of this study.

ACKNOWLEDGMENTS

The authors thank all the study participants and also all the people that helped by suggesting organizations to participate in the study or spreading the study call for participation.

REFERENCES

[1] Monalessa Perini Barcellos. 2020. Towards a Framework for Continuous Software Engineering. In *Proceedings of the 34th Brazilian Symposium on Software Engineering* (Natal, Brazil) *(SBES '20)*. Association for Computing Machinery, New York, NY, USA, 626–631. https://doi.org/10.1145/3422392.3422469

[2] Kent Beck. 2000. *Extreme programming explained: embrace change*. addison-wesley professional.

[3] Jan Bosch. 2014. Continuous Software Engineering: An Introduction. In *Continuous Software Engineering*. Springer, Chapter 1, 3–13.

[4] Patrick Debois et al. 2011. Devops: A software revolution in the making. *Journal of Information Technology Management* 24, 8 (2011), 3–39.

[5] Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. 2008. *Selecting Empirical Methods for Software Engineering Research*. Springer London, London, 285–311. https://doi.org/10.1007/978-1-84800-044-5_11

[6] Brian Fitzgerald and Klaas-Jan Stol. 2017. Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software* 123 (2017), 176–189. https://doi.org/10.1016/j.jss.2015.06.063

[7] Jan Ole Johanssen, Anja Kleebaum, Barbara Paech, and Bernd Bruegge. 2018. Practitioners' Eye on Continuous Software Engineering: An Interview Study. In *Proceedings of the 2018 ICPS* (Gothenburg, Sweden) *(ICSSP '18)*. Association for Computing Machinery, New York, NY, USA, 41–50. https://doi.org/10.1145/3202710.3203150

[8] Teemu Karvonen and et. al. 2016. The CRUSOE Framework: A Holistic Approach to Analysing Prerequisites for Continuous Software Engineering. In *Product-Focused Software Process Improvement*. Springer International Publishing, Cham, 643–661.

[9] Teemu Karvonen, Lucy E. Lwakatare, Tanja Sauvola, Jan Bosch, Helena H. Olsson, Pasi Kuvaja, and Markku Oivo. 2015. Hitting the target: practices for moving toward innovation experiment systems. In *Int. Conference on Software Business*. 117–131.

[10] Leonardo Leite, Carla Rocha, Fabio Kon, Dejan Milojicic, and Paulo Meirelles. 2019. A Survey of DevOps Concepts and Challenges. *ACM Comput. Surv.* 52, 6, Article 127 (nov 2019), 35 pages. https://doi.org/10.1145/3359981

[11] Leann Myers and Maria J Sirois. 2004. Spearman correlation coefficients, differences between. *Encyclopedia of statistical sciences* 12 (2004).

[12] Helena Holmstrom Olsson, Hiva Alahyari, and Jan Bosch. 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*. 392–399.

[13] Per Runeson, Martin Host, Austen Rainer, and Bjorn Regnell. 2012. *Case study research in software engineering: Guidelines and examples*. John Wiley & Sons.

[14] Paulo Sérgio dos Santos, Monalessa P. Barcellos, and Rodrigo Fernandes Calhau. 2020. Am I Going to Heaven? First Step Climbing the Stairway to Heaven Model Results from a Case Study in Industry *(SBES '20)*. ACM, 309–318. https://doi.org/10.1145/3422392.3422406

[15] Paulo S. dos Santos Jr, Monalessa P. Barcellos, Fabiano Borges Ruy, and Moisés S. Omêna. 2022. *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

[16] Paulo S dos Santos Jr, Monalessa Perini Barcellos, and Fabiano Borges Ruy. 2021. Tell Me: Am I Going to Heaven? A Diagnosis Instrument of Continuous Software Engineering Practices Adoption. In *Evaluation and Assessment in Software Engineering* (Trondheim, Norway) *(EASE 2021)*. Association for Computing Machinery, New York, NY, USA, 30–39. https://doi.org/10.1145/3463274.3463324

[17] Mojtaba Shahin, Muhammad Ali Babar, and Liming Zhu. 2017. Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices. *CoRR* (2017). http://arxiv.org/abs/1703.07019

[18] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in software engineering*. Springer Science & Business Media.