Tell me: Am I going to Heaven? A Diagnosis Instrument of Continuous Software Engineering Practices Adoption

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

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

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

ABSTRACT

Context: Continuous Software Engineering (CSE) involves a set of practices that aims at making software development continuous and integrated to business. However, moving from traditional to integrated, agile and data-driven software development requires changes in the organizations culture, practices and structure, which may not be easy. *Objective*: Our focus is to help organizations get an overall view of the CSE practices they perform, identify where they are in the CSE evolutionary path and which areas should be improved. *Method*: We created a diagnosis instrument, called *Zeppelin*, to evaluate the adoption of CSE practices and applied it in five Brazilian software organizations. *Results*: Zeppelin was recognized as a useful tool to help organizations have a more comprehensive view of their CSE practices and envision the evolution and improvement path they can follow. *Conclusion*: Zeppelin supports software development organizations to get a big picture of CSE practices and identify their position in the CSE evolution path. Moreover, it contributes to identify strategies to advance software development towards a CSE environment.

CCS CONCEPTS

• **Software and its engineering** → *Software creation and management*; **Software development process management**;

KEYWORDS

Stairway to Heaven, Diagnosis, Continuous Software Engineering, Agile

ACM Reference Format:
Paulo Sérgio dos Santos Júnior, 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 (EASE 2021), June 21–23, 2021, Trondheim, Norway.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3463274.3463324

1 INTRODUCTION

Characteristics and demands of the modern and digital society have transformed the software development scenario and presented new challenges to software developers and engineers, such as the need for faster deliveries, frequent changes in requirements, lower tolerance to failures, and the need to adapt to contemporary business models. The adoption of agile practices has allowed organizations to shorten development cycles and increase customer collaboration. However, this has not been enough. Continuous actions of planning, construction, operation, deployment, and evaluation are necessary to produce products that properly meet customers' needs, to make well-informed decisions and identify business opportunities. Thus, organizations should evolve from traditional to continuous and data-driven development in a continuous software engineering approach [3].

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 the business. It understands that the development process is not a sequence of discrete activities, performed by distinct and disconnected teams. It aims to establish a continuous flow between software-related activities, taking into account the entire software life cycle. It is a recent topic that seeks to transform discrete development practices into more iterative, flexible and continuous alternatives, keeping the goal of building and delivering quality products according to established time and costs [8].

Considering that organizations struggle with the changes to be made to implement CSE practices, Olsson et al. [18] proposed the Stairway to Heaven model (StH), which describes the typical successful evolution path of an organization from traditional to customer data-driven development. The model comprises five stages: Traditional Development, Agile Organization, Continuous Integration, Continuous Deployment, and R&D as an Innovation System. Complementarily, Karvonen et al. [14] proposed an extension to StH by indicating key practices to move from a stage to another. However, these works address StH stages at a high level, not detailing their practices or supporting to identify at which stage (or stages) an organization is. This lack of guidance makes it difficult for an organization to identify at which stage(s) it is, and which practices it could perform or improve to advance in the CSE evolutionary path.

EASE 2021, June 21–23, 2021, Trondheim, Norway

ACM ISBN 978-1-4503-9053-8/21/06…$15.00
https://doi.org/10.1145/3463274.3463324

Some maturity models, such as CMMI [5] and MR-MPS-SW [6][22] and standards such as ISO/IEC 15504 [11] help organizations improve software processes by defining practices and results organized into levels. Thus, organizations are evaluated and positioned according to their processes' maturity and capacity and are able to improve their processes by aiming at the following levels. However, these models adopt deterministic and normative software process improvement. In CSE context, organizations need to be more flexible, encourage innovation and self-organization and promote sustainable development processes [1][8][18][21]. In this sense, the StH model provides a simpler and pragmatic view of how organizations evolve their process towards CSE. Although StH is organized in five stages, due to the agile and flexible environment, organizations may perform practices from different stages while evolving from traditional to continuous and data-driven development [14].

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, we propose a diagnosis instrument called **Zeppelin**[1]. It helps identify the degree of adoption of CSE practices in each StH stage. 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 areas should be improved. Thus, it is possible to develop a plan to improve and advance software development towards CSE. Zeppelin uses StH [18] as reference model and also considers Continuous* activities proposed in [8], CSE practices and aspects provided in the Eye of CSE [12] and CSE processes constituting the CSE framework (hereafter, called FCSE) proposed in [3].

This paper contributes to the state of the art by consolidating knowledge from several CSE frameworks ([3], [8], [12], and [18]) and by extending knowledge about StH stages adding to them practices suggested in these frameworks. It also gives a contribution to the of state of the practice, because Zeppelin can be used by software engineers to diagnose the adoption of CSE practices in organizations and, thus, support the definition of strategies to improve software development.

We have applied Zeppelin in five Brazilian software organizations to evaluate it and understand how CSE practices have been applied, how these organization are positioned in relation to StH stages and which areas they should address in improvement actions. The study was performed between September/2020 and February/2021. As results, the study's participants considered Zeppelin useful. They agreed that it provided a faithful picture of CSE practices in the organization and that the information provided is useful to create strategies to evolve software development efforts.

This paper introduces Zeppelin and presents the main results of its use in those five organizations. It is organized as follows: Section 2 presents the theoretical background; Section 3 introduces Zeppelin; Section 4 presents the study planning, execution, results, and threats to validity; and Section 5 presents our 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. The '*big picture*' by which this might be achieved goes beyond agile principles and surfaces a more holistic set of continuous activities [8]. According to Johanssen et al. [12], 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.

In the last years, some works have addressed CSE processes and practices. Four of them are particularly relevant to this work: [3], [8], [12], and [18]. Olsson et al. [18] defined the Stairway to Heaven model (StH), 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*. As the organization starts showing the benefits of working agile, system integration and verification become involved and continuous integration is adopted. Once *continuous integration* runs internally, lead customers often express an interest to receive software functionality earlier than through the normal release cycle. They want *continuous deployment* of software. The final stage is *R&D as innovation system*, 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.

Fitzgerald and Stol [8] 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. They introduce BizDev, analogous to DevOps, but referring to the continuity and alignment between business strategy and software development.

From interviews performed with CSE practitioners, Johanssen et al. [12] defined the Eye of CSE, consisting 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. The proposal differs from the sequential nature of the StH model [18] as the authors argue that even if some CSE elements, such as continuous integration and delivery, require a stepwise introduction, CSE should be approached from multiple angles simultaneously.

[1] The name Zeppelin was chosen because the diagnosis 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.

Barcellos [3] proposes a framework (FCSE) containing a set of 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 [8], elements from the Eye of CSE [12] and StH stages were considered to define FCSE [18]. Differently from StH, the framework considers that processes can be performed simultaneously and gradually.

In the literature, there are several proposals to evaluate software process (e.g., [9], [16], [17], and [7]). However, these works are not focused on CSE. The study most closely related to our work was conducted by Karvonen et al. [15]. It introduces the CRUSOE (Continuous inteRdependencies in prodUct-focused SOftware Engineering) framework, which allows analyzing CSE prerequisites in software-intensive projects (e.g., embedded systems), based on interdependence among ecosystem, strategy, architecture and organization dimensions[15]. The prerequisites enable to create strategies to move a project from traditional product development to CSE. Different from Zeppelin, CRUSOE uses 14 questions to identify the interdependence among the aforementioned dimensions and focuses on creating strategies to align business and development using the identified interdependence. In addition, CRUSOE does not identify at which degree CSE practices are adopted in an organization. Zeppelin aids in identifying the degree of adoption of CSE practices in an organization, how it positions in relation to StH stages and which areas can be improved to advance in the CSE evolutionary path. Furthermore, Zeppelin provides an analytic report that summarizes information about CSE practices adoption and enables to visualize in an easier way strengths and weaknesses related to StH stages and nine dimensions defined based on [3], [8], and [12]. In summary, CRUSOE seeks to identify the relationships among business, strategies and architecture dimensions, while Zeppelin focuses on the adoption degree of CSE practices in an organization and on providing a holistic view about CSE adoption in an organization, without concern with the dimensions explored in CRUSOE.

3 ZEPPELIN: A DIAGNOSTIC INSTRUMENT FOR CSE

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 culture and analyzing artifacts, processes, tools, people and other elements present in software development [20]. 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, we created **Zeppelin**.

The work followed the Design Science Research (DSR) paradigm, which concerns extending human and organizational capabilities by creating new and innovative artifacts [10][2]. We used this research approach because the object of study is an artifact in context — specifically, a diagnosis instrument that identifies CSE practices adopted in an organization — and its evaluation was performed in five real organizational environments. DSR is an iterative process including three related cycles. In short, in *Relevance Cycle*, we identified the problem to be addressed, its motivations and requirements that the proposed artifact should meet to solve the problem. In the *Design Cycle*, we developed and evaluated the artifact (i.e., Zeppelin). For reaching Zeppelin current version, we performed four cycles of design and evaluation activities. In each cycle, the first author designed Zeppelin, and the second and third authors evaluated it using peer review technique and suggested improvements. These evaluations aimed at increasing the instrument quality, by verifying its adherence to CSE and fixing problems that could lead to misunderstandings. The resulting diagnosis instrument was then applied in five organizations. In the *Rigor Cycle*, which refers to using and generating knowledge, we used knowledge mainly about CSE, process evaluation and case study. Our main contribution is Zeppelin, which provides means to understand CSE practices adoption in an organization and identify directions to improve software development.

Zeppelin has two components: *Diagnosis Questionnaire*, which identifies the CSE practices an organization performs and the degree to which they are adopted; and *Analytics 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. Next, we provide details about each component.

3.1 Diagnosis Questionnaire

The *Diagnosis Questionnaire* consists of an electronic spreadsheet with eight forms: *Context*, to provide a brief introduction to CSE; *Instructions*, which guides the user on how to fill in the other forms; *Organization*, to characterize the organization (e.g., organization type, size, age, development team size); *User*, to characterize the person answering the questionnaire on the organization behalf (e.g., position, knowledge and experience with CSE practices); and four forms concerning StH stages and containing in all 76 statements expressing CSE practices: *Agile Organization* (26 practices), *Continuous Integration* (18 practices), *Continuous Deployment* (19 practices) and *R&D as Innovation System* (13 practices).

CSE practices were identified based on the literature and (mainly on [3], [8], [12] and [18]) and on the authors practical experience. For example, we identified some practices related to process from [3] and [8] (e.g., Agile Development, Business Alignment, Continuous Integration, Continuous Deployment, Continuous Software Measurement, Continuous Knowledge Management, Continuous Quality Assurance, among others). Besides being organized in stages, practices are also associated to categories (called dimensions in Zeppelin) and elements extracted from the Eye of CSE (e.g., Development, Quality, Software Management, Team, Technical Solution, Knowledge, Operation, Business, and User/Customer). For example, the statement "*The project team has autonomy to make technical decisions on the project*" is related to the Agile Organization stage and to the Software Management dimension. "*Tests run automatically, periodically, in a test environment, to verify code coverage*" is related to the Continuous Integration stage and to the Development dimension. "*The continuous deployment process is automated*" relates to the Continuous Development stage and to the Software Management dimension. "*Experiments (e.g., A/B tests) are conducted with* *customers/consumers to improve products.*" is associated to the R&D as Innovation System stage and to the User/Customer dimension.

When applying Zeppelin, for each statement, the user must indicate the level to which the referred CSE practice is adopted in the organization. This way, besides knowing how many practices have been adopted, the organization can understand how comprehensive their adoption has been. The adoption levels were defined based on [14] 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 the organization has never used. *Abandoned* level represents practices that were discontinued. Based on this information, the software engineer can investigate barriers that have prevented the referred practice implementation or opportunities to implement it. For example, she/he might identify that daily meetings were abandoned because the development team's members work geographically distributed and in different time zones. Thus, she/he can consider other methods to address communication issues. *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 established 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. Figure 1 shows the form used to evaluate practices related to the *Continuous Integration* stage.

Table 1 presents some statements contained in the forms related to the other stages. Due to space limitation, in this paper we show only a fragment of the *Diagnosis Questionnaire*. The complete *Diagnosis Questionnaire* is available at [13].

3.2 Analytics Report

The *Analytics Report* is an artifact that imports data from the *Diagnosis Questionnaire*, consolidates it and provides a panoramic view (by using tables, charts and text) of CSE practices adoption in the organization. It focuses on answering three main questions, which provide different perspectives of CSE practices in the organization:

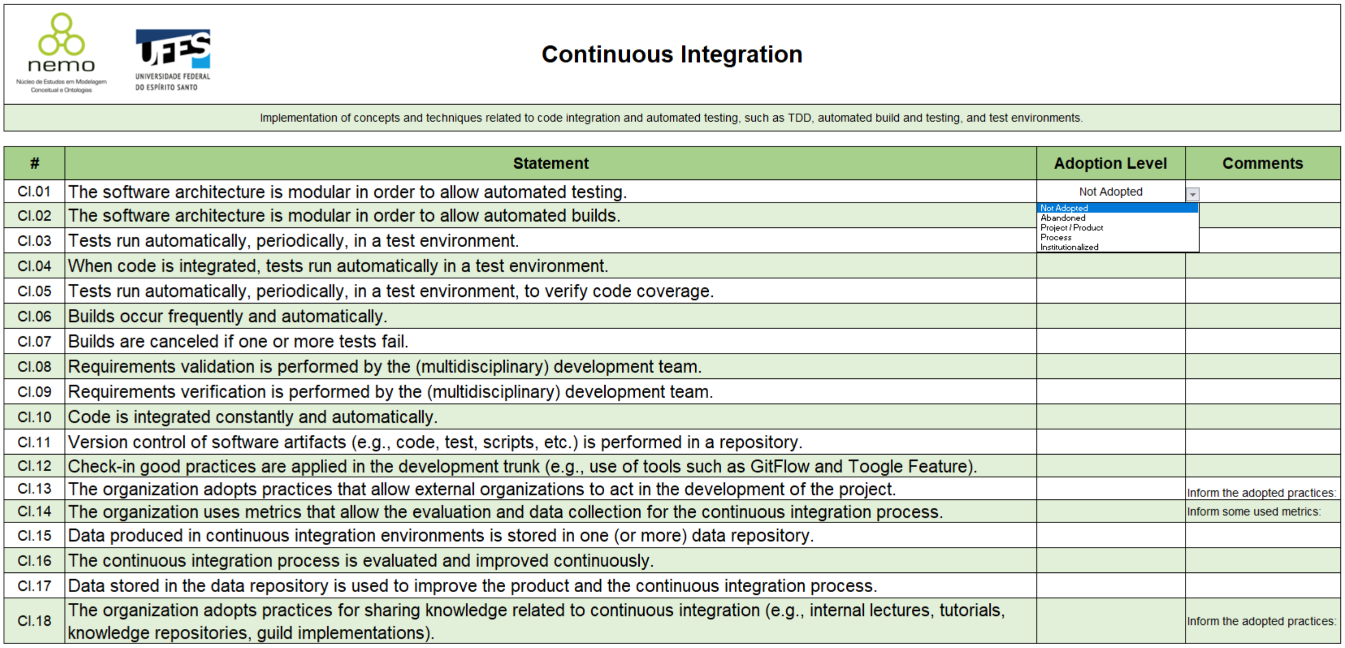
Q1) *How much have CSE practices been adopted at each StH stage?* The answer for this question is a high-level view of the adoption of CSE practices in the organization. Zeppelin provides a summarized table and a chart, showing the degree of adoption of CSE practices at each StH stage (e.g., an organization may have achieved 70% of the *Agile Organization* stage). Thus, the organization can see how much it addresses practices related to each StH stage. This view allows observing if the organization is at multiple stages, with different degrees of adoption. The degree of adoption at each stage (DA) 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, DAstage = (weightALpractice1 +...... + weightALpracticen) / n) * 100). The weights for the adoption levels vary from 0 (zero) (referring to the *Not Adopted* level) to 1.0 (referring to the *Institutionalized* level). The answer for this question helps the organization find out which stages it has covered, and which ones need improvements.

Q2) *Taking dimensions defined in the Eye of CSE as a reference, how much have CSE practices been adopted at each StH stage*? Within this perspective, the organization can learn dimensions it has been more (or less) focused on. For example, an organization could find out that it is doing very well in CSE practices related to *Technical Solution*, but it should improve the ones related to Business.

Q3) *Taking elements defined in the Eye of CSE as reference, how much have CSE practices been adopted at each StH stage*? This question aims at detailing the answer to Q2 by showing the

Table 1: Examples of CSE practices related to StH stages defined in the Diagnosis Questionnaire

Stage of StH	Statement	
Agile Organization	AO.02	Project teams include a product owner who is responsible for representing the Customer and actively participates in the projects.
	AO.03	The scope of the project is defined gradually, using the Product Backlog (or equivalent artifact).
	AO.15	The project team has autonomy to make technical decisions on the project.
	AO.17	Good programming practices are adopted (e.g., collective coding, standardized coding, pair programming, code review, etc.).
Continuous Deployment	CD.01	The main customers/consumers are identified and participate in the development process, influencing the functionalities that will be produced and delivered.
	CD.02	There is a clear flow of information between Development and Operation, allowing that new functionality developed to go live automatically.
	CD.16	Data produced in continuous deployment environments is stored in one (or more) data repository.
R&D as Innovation System	IS.02	Feedbacks (data and opinions) from customers / consumers are continuously and automatically captured.
	IS.05	Feedbacks (data and opinions) from customers / consumers are used for experimentation and innovation.
	IS.09	The organization has a clear 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.


nemo
Núcleo de Estudos em Modelagem Conceitual e Ontologias
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
Not Adopted
Abandoned
Project / Product
Process
Institutionalized

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
CI.01	The software architecture is modular in order to allow automated testing.	Not Adopted	
CI.02	The software architecture is modular in order to allow automated builds.		
CI.03	Tests run automatically, periodically, in a test environment.		
CI.04	When code is integrated, tests run automatically in a test environment.		
CI.05	Tests run automatically, periodically, in a test environment, to verify code coverage.		
CI.06	Builds occur frequently and automatically.		
CI.07	Builds are canceled if one or more tests fail.		
CI.08	Requirements validation is performed by the (multidisciplinary) development team.		
CI.09	Requirements verification is performed by the (multidisciplinary) development team.		
CI.10	Code is integrated constantly and automatically.		
CI.11	Version control of software artifacts (e.g., code, test, scripts, etc.) is performed in a repository.		
CI.12	Check-in good practices are applied in the development trunk (e.g., use of tools such as GitFlow and Toogle Feature).		
CI.13	The organization adopts practices that allow external organizations to act in the development of the project.		Inform the adopted practices:
CI.14	The organization uses metrics that allow the evaluation and data collection for the continuous integration process.		Inform some used metrics:
CI.15	Data produced in continuous integration environments is stored in one (or more) data repository.		
CI.16	The continuous integration process is evaluated and improved continuously.		
CI.17	Data stored in the data repository is used to improve the product and the continuous integration process.		
CI.18	The organization adopts practices for sharing knowledge related to continuous integration (e.g., internal lectures, tutorials, knowledge repositories, guild implementations).		Inform the adopted practices:

Figure 1: Fragment of the Diagnosis Questionnaire with the 18 statements related to Continuous Integration.

elements (related to each dimension) the organization have developed well and the ones that need improvements. For example, looking closer at *Technical Solution*, an organization could visualize that it reaches 50% of *Code Review* practices and 90% of *Version Control* practices.

With the *Analytics Report* in hands, one can analyze the organization starting from a higher-level view (Q1), understanding how CSE practices are distributed and adopted at each StH stage. Then, he/she can drill-down this view by observing, from a dimension perspective (Q2), dimensions in which CSE practices have been more advanced and which ones need more efforts. Finally, this view can be detailed by the element perspective (Q3), revealing strengths and weaknesses inside each dimension . An electronic spreadsheet containing examples of tables and charts provided in the *Analytics Report* is available at [13].

4 APPLYING THE DIAGNOSIS INSTRUMENT

In this section, we present the study carried out to evaluate Zeppelin. Section 4.1 presents the study goal, research questions and participants. Section 4.2 addresses data collection. Section 4.2 regards analysis and interpretation. Last, in Section 4.4, we discuss study limitations and threats to the validity.

4.1 Study Design

The study *goal* was to evaluate whether Zeppelin is useful for organizations to understand their position in relation to CSE practices and envision improvements to evolve software development towards CSE. Aligned to this goal, two research questions were defined: (RQ1) *Is Zeppelin useful to identify which CSE practices have been adopted in an organization and provide a panorama about its position in the CSE evolutionary path*? (RQ2) *Does Zeppelin help an organization envision an improvement path to follow*?

The study involved five Brazilian organizations (here called Org1 to Org5 for anonymity reasons). Org1 is a startup which provides a solution for organizations working with industry 4.0. It has five developers and one tech lead, and has performed CSE practices since its creation, in 2019. It has a development team composed of all developers and the tech lead.

Org2 is also a startup and provides solutions for waste management. It has 15 developers and four engineers. It was founded in 2018, and it has performed CSE practices since August/2019. The developers and engineers are divided in three teams. Each team is responsible for a product. A single tech lead is responsible for defining technology, software architecture and managing products; one Requirement Engineer is responsible for requirements elicitation.

Org3 is an organization which develops software for other organizations (i.e., a software house). It has 40 developers, was founded in 2010, and has performed CSE practices since 2019. Its development team is composed of developers and a tech lead. The tech lead is responsible for requirements elicitation and project management, while the developers build the product.

Org4 is a fintech with 40 developers. It was founded in 2000. Each product has a dedicated development team composed of product owner, developers, designers, and a customer analyst. Each development team works as a business unity responsible for maintaining and managing the product financially.

Finally, Org5 is a Brazilian public organization responsible for supervising the processes of justice. It has 30 developers, building and maintaining its software. It has performed CSE practices since 2019.

Table 2: Participants and organizations involved in study.

Participant	Role	Organization Type	Experience with CSE practices
P1	Product Owner	Startup	1 to 3 years
P2	Software Engineering	Startup	1 to 3 years
P3	Head of Software Engineering	Software House	3 to 5 years
P4	Head of Software Engineering	Fintech	3 to 5 years
P5	Software Engineering	Public Organization	1 to 3 years

The study *participants* were five employees (one of each organization) in charge of managing software development. Here we called them P1 to P5. P1 is a Product Owner and works at Org1 since it was founded. P2 is a Software Engineer and works at Org2 since its foundation. P3 is a Head of Software Development and works at Org3 since 2010. P4 is also a Head of Software Development and works at Org4 since 2017. P5 is a Software Developer (he also performs management activities) and works at Org5 since 2014. Table 2 summarizes information about the participants and respective organizations.

The *procedure* adopted in the study consisted of five steps: (i) the authors sent the Diagnosis Questionnaire to the participants; (ii) the participants filled in the Diagnosis Questionnaire and sent it back to the authors; (iii) the first and third authors performed an interview with the participants to confirm her/his answers and get detailed information; (iv) the authors created the Analytics Report related to each organization, with some complementary information, and sent it to the participants; and, finally (v) the participants filled in a feedback form to provide their opinion about Zeppelin.

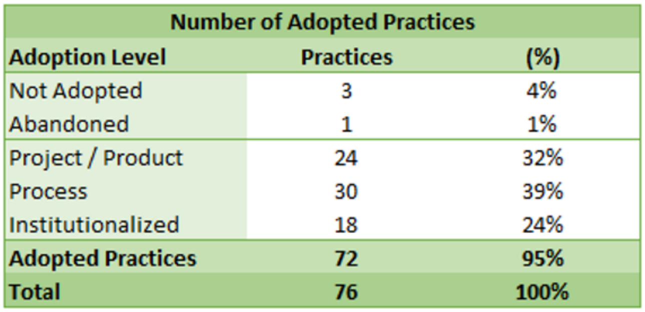
4.2 Study execution and data collection

The study was performed between September/2020 and February/2021, by following the described procedure above. After each participant filed in the Diagnosis Questionnaire, we performed an interview. We started with general questions to gather information about the organization (e.g., age, number of employees, number of developers and when it started to perform CSE practices) and interviewee's knowledge of and experience with CSE practices. Then, for each statement of the Diagnosis Questionnaire, we asked the participant to explain how the practice has been adopted in the organization, so that we could confirm the answer she/he gave in the questionnaire and obtain additional information to better understand how the organization works. The participants were told to feel free to talk as much as they wanted to.

Each interview lasted about 60 minutes and it was divided in two parts. The first part was used for the interviewee to give more details about his/her answers to the Diagnosis Questionnaire, confirming (or changing) them, and for the interviewers to get more information about the organization and how it has performed CSE practices. In the second part, the interviewers presented some tables and charts automatically generated in a spreadsheet using the answers to the Diagnosis Questionnaire to provide to the interviewee a panoramic view of the adoption of CSE practices in the organization and verify if he/she found the view faithful to the organization reality. These tables and charts were latter used to create the Analytics Report.

After data validation through the interviews, we elaborated the Analytics Report and sent it to the participants. The report contained tables and charts related to question Q1-Q3 (see Section 3.2), information summarizing data from each table/chart, and a conclusion about the adoption of CSE practices in the organization plus a reflection about possible improvements. An example of Analytics Report delivered to the participants can be seen at [13]. Next, we present some extracts of the *Analytics Report* referring to Org4. Some perceptions obtained from data provided by the participants in the Diagnosis Questionnaire and consolidated in the Analytics Reports are discussed in Section 4.2.

Figure 2 shows the number of practices Org4 adopts. It can be seen that Org 4 has a good coverage of CSE practices, performing, with different adoption levels, 72 out of 76 CSE practices, i.e., a rate of 95%.

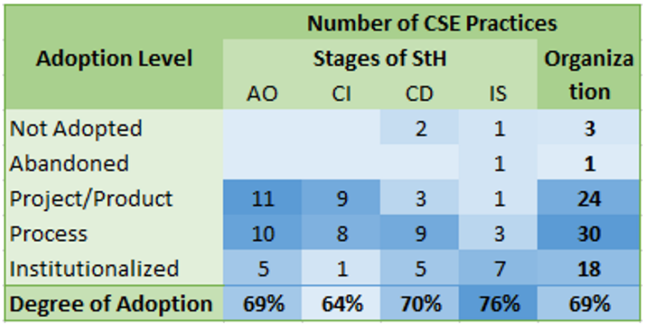


Number of Adopted Practices		
Adoption Level	**Practices**	**(%)**
Not Adopted	3	4%
Abandoned	1	1%
Project / Product	24	32%
Process	30	39%
Institutionalized	18	24%
Adopted Practices	**72**	**95%**
Total	**76**	**100%**

Figure 2: Fragment of the Diagnosis Questionnaire

Figure 3 illustrates a table related to Q1 (StH stage perspective). It shows how many practices related to each StH stage Org4 has performed and at each adoption level, giving a panoramic view of the CSE practices distribution. Based on that, the degree of adoption at each stage is calculated. In the figure, different shades of blue are used according to the cell value (darker shades are used to highlight higher values).

As it can be seen, in Org4, most of the practices are performed at *Process* (30 CSE practices) and *Product/Project* level (24 CSE practices). There are also many *Institutionalized* practices (18). There are only three *Not Adopted practices* and one *Abandoned* practice. When considering the number of adopted practices and also the respective levels of adoption, Org 4 reaches a total degree of adoption of 69%. The degree of adoption (69%) is smaller than the rate of performed CSE practices (95%) illustrated in Figure 3 because many practices are performed only in the context of some projects or products instead of in the organization as a whole. By knowing both, rate of CSE performed practices and degree of adoption, it is possible better understand how many practices the organization has adopted and how comprehensive they have been. When looking



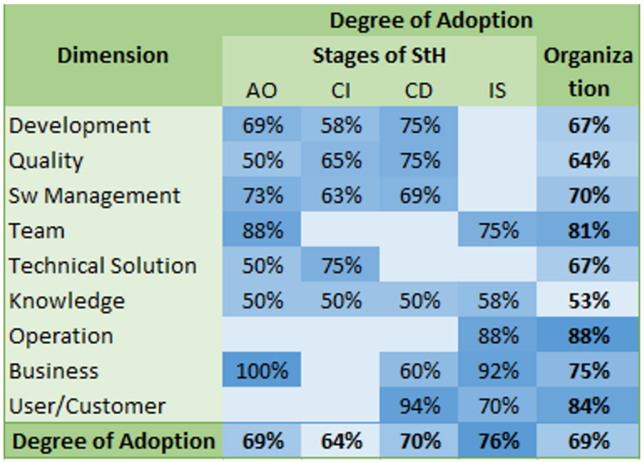
Adoption Level	Number of CSE Practices				
	Stages of StH				Organiza tion
	AO	CI	CD	IS	
Not Adopted			2	1	3
Abandoned				1	1
Project/Product	11	9	3	1	24
Process	10	8	9	3	30
Institutionalized	5	1	5	7	18
Degree of Adoption	**69%**	**64%**	**70%**	**76%**	**69%**

Figure 3: Number of practices adopted in Org4.

at StH stages, we also note a good coverage of CSE practices, but the adoption degree is higher in the Continuous Deployment (70%) and R&D as Innovation System (76%) stages (mainly because at these stages, there are more practices in the higher levels of adoption).

P4 reported that Org4 has an enterprise architecture (e.g., processes, culture, and IT infrastructure) that promotes the agile and innovation culture and alignment among *Business*, *Development* and *Operation*. According to him, each development team is self-organized, and it is responsible for building, maintaining one or more products; the Product Owner is an expert in the product domain; and each development team has a Customer Success Analyst which is responsible for collecting and analyzing customer feedback (data and opinion) in order to support product improvement. P4 also informed that Test A/B was momentarily abandoned, because the development teams do not have had resources and time to perform it.

Figure 4 illustrates results related to Q2, showing the adoption of CSE practices in Org 4 from the dimension perspective. The degrees of adoption were calculated considering the number of CSE practices related to each dimension and their level of adoption. As shown in Figure 4, Org4 has a higher degree of adoption in CSE practices related to *Operation* (88%), *User/Customer* (84%), and *Team* (81%). Looking at each StH stage, the higher degrees of adoption refer to practices related to Business (100%) at Agile Organization stage, *Technical Solution* (75%) at Continuous Integration, *User/Customer* (94%) at Continuous Deployment, and *Business* (92%) at R&D as Innovation System. Concerning the *Technical Solution* category, P4 said that Org4 invested in automating most of the CSE practices related to the Continuous Integration stage to assure the quality of the software artifacts. For example, the choice of the reviewer during the *code review* process is automated and a reviewer only is indicated when the committed code passes all automated tests. Otherwise, the CI engine sends a message to the developer reporting code errors. Regarding the CSE practices related to *Business* category at R&D as Innovation System, P4 pointed out that Org4 has a data science team that works together with the Customer Success Analysts to understand the customer feedback data, and thus, provides information to support decision-making in different levels of Org4. in Figure 5 shows a radar chart as a complementary representation of data presented in the table showed in Figure 4.



Dimension	Degree of Adoption				
	Stages of StH				Organiza tion
	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%**

Figure 4: Adoption of CSE practices per dimension at each StH stage and adoption level in Org4.

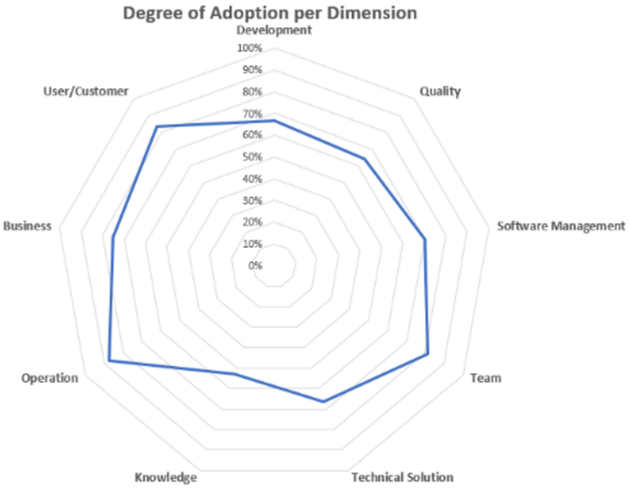


Figure 5: Adoption of CSE practices at each dimension in Org4.

As it can be seen in Figure 5, Org4 has focused less on CSE practices related to Knowledge than to the other dimensions. P4 commented that Org4 is aware that needs to improve practices related to knowledge, such as knowledge capture, storage and reuse, and that it intends to adopt new practices in this context. According to P4, the lack of adequate knowledge management brings problems to different parts of the software development process.

Figure 6 shows a table related to Q3. It details the dimension view by presenting the adoption of CSE practices according to the elements they relate to. By looking at Figure 6, it is possible to identify the degree of adoption associated to each element. This helps identify weakness and strengths at a lower level. For example, in the *Team* dimension, it is possible to notice that *Self-reflection and discipline* is the element with highest degree of adoption (100%) and represents a strength of Org4. On the other hand, when analyzing the *Development* dimension, it is noted that *Continuous*

Requirements Engineering probably represents a weakness of Org4, since it has the lowest degree of adoption (50%).

After receiving the Analytics Report, the participants were asked to share their perceptions about the use of Zeppelin by filling in a feedback form. The feedback form is composed of nine questions distributed in three sections. The first section contains six questions that collect data about how useful Zeppelin was to provide a panoramic view of CSE practices adoption in the organization and help identify areas to be addressed in improvement actions. The options of answer follow the Likert Scale. The questions in this section were the following: (i) *Did Zeppelin provide a broad view of practices related to CSE in your organization*? (ii) *Did Zeppelin support to identify aspects your organization needs to improve*? (iii) *Did Zeppelin provide a faithful panoramic view of your organization*? (iv) *Did Zeppelin help you identify weakness of your organization*? (v) *Did Zeppelin help you identify strengths of your organization*? (vi) *Was Zeppelin useful for your organization to define strategies to improve software development*?

The second section contains two questions used to capture information to improve Zeppelin: (i) *Is there any information of the diagnosis result that is not in accordance with the reality of the organization or was not captured*? (ii) *Do you have any additional comment or suggestion for Zeppelin improvement*? If the participant answered "yes" to any of these questions, he/she was free to provide feedback using their own words in a text field.

The last section of the feedback form aims to identify which CSE dimensions the organizations intend to focus on improvement actions. Our goal was to verify what dimensions the organizations consider more important to improve, based on data provided in the Analytics Report. This section contains a single question: *Which dimensions do you consider most important for your organization to improve in the CSE evolution path*?

4.3 Study analytics and interpretation

This section analyses data obtained in the study. First, we make a general discussion about data provided by the participants in the Diagnosis Questionnaire. Our goal is to provide an overview of how CSE practices have been adopted in the studied organizations. Then, aiming to answer the research questions and improve Zeppelin, we analyze the participants perceptions about Zeppelin, based on their answers to the feedback form.

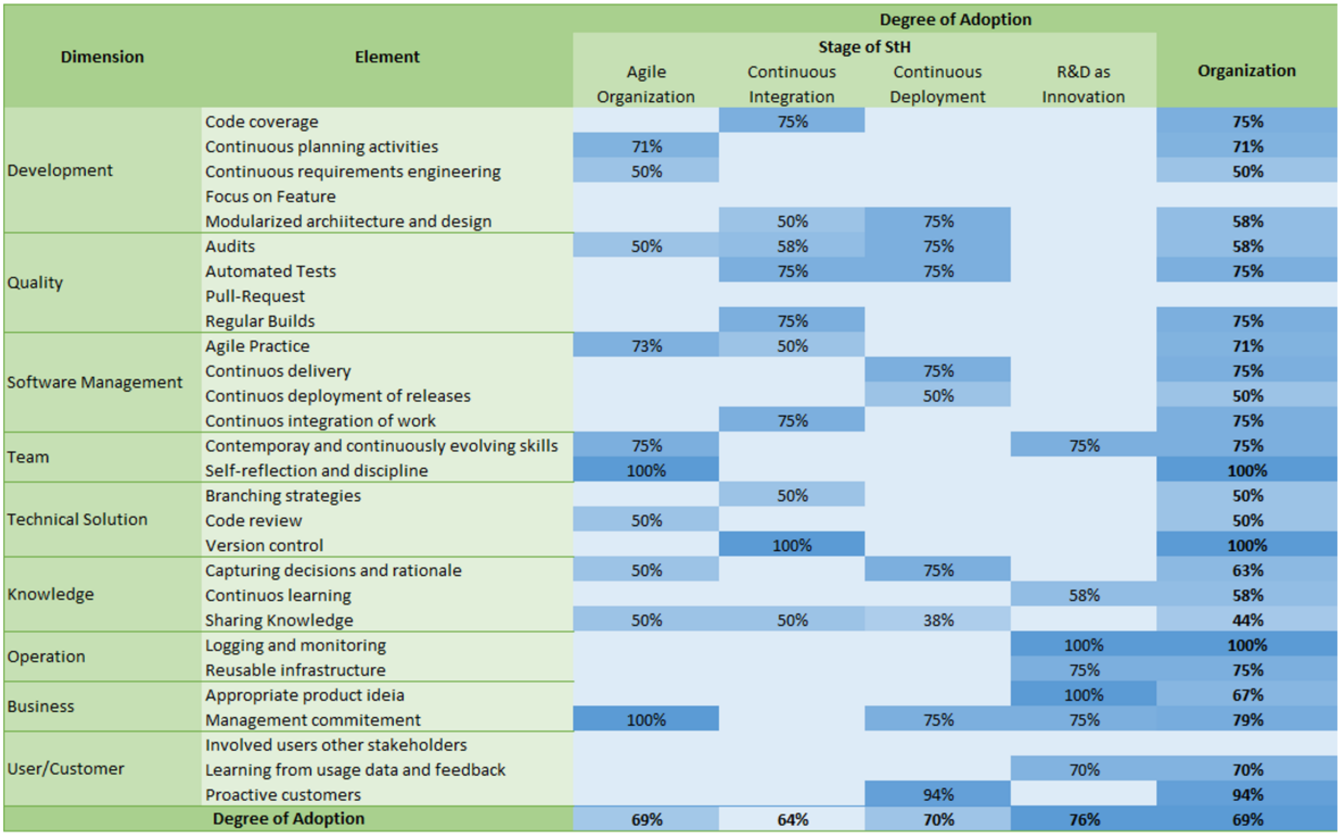
Analyzing the results from the Analytics Reports considering data from all organizations involved in the study, we noticed that most of the practices are adopted at *Product/Project* level (43%) or are *Not Adopted* (35%). 13% of the practices are *Institutionalized.* 8% are adopted at *Process level* and only 1% were *Abandoned.* Concerning StH stages, the higher degrees of adoption occur in the *Agile Organization* (51%) and *Continuous Integration* (39%) stages. Moreover, the higher rates of performed activities also occur in these same stages (82% at *Agile Organization* and 71% at *Continuous Integration*). These results indicate that most of the organizations involved in the study have started to perform CSE practices in the project or product context and did not evolve them enough yet to reach the organization as a whole. In fact, as argued by Olsson et al. [18], organizations often start performing CSE practices in some projects and when they learn how to perform the practices and note good results, they evolve to apply CSE practices in the organizational context.

Regarding dimensions, the higher degrees of adoption occur in *Technical Solution* (51%), *User/Customer* (34%), and *Software Management* (37%). Considering StH stages, practices related to *Software Management* (43%) and *Team* (41%) have a higher degree of adoption in *Agile Organization* stage; *Technical Solution* (61%) and *Software Management* (27%) in Continuous Integration stage; *User/Customer* (44%) and *Business* (30%) in Continuous Deployment, and *Business* (28%) and Knowledge (28%) in R&D as Innovation System stage. These results reveal that in most of the subject organizations involved in the study the first stages of StH (*Agile Organization* and *Continuous Integration*) have been dedicated to developing hard skills and soft skills needed in the software development process, while the last stages (*Continuous Deployment* and *R&D as Innovation System*) have been used to promote innovation and improve the product. For example, Org1, Org2, Org3 and Org5 are discovering how to implement or adapt CSE practices (e.g., code review, automated test, communication, agile culture) in their software development process, while Org4 has used data obtained from CSE practices related to Continuous Deployment and R&D as System Innovation (*Business*-100% and *Continuous Learning*-58%) to understand the clients and improve its products.

When comparing results from all the organizations, we noticed that even organizations of the same type have adopted different CSE practices, according to their needs and particularities. For example, although Org1 and Org2 are startups, each of them adopts different CSE practices. We also noticed that the organization size, age, and maturity influence the adoption of CSE practices. For example, Org4 is the biggest, oldest and more mature organization of the study, and its results were better than the ones of the other organizations (i.e., it adopts more practices and at a higher degree).

To answer the study research questions, we considered the participants' feedback. Concerning (RQ1) *Is Zeppelin useful to identify which CSE practices have been adopted in an organization and provide a panorama about its position in the CSE evolutionary path*?, we analyzed the answers to the questions of the first section of the feedback form. As result, all participants agreed (80% strongly agreed and 20% agreed) that Zeppelin provided a comprehensive view of CSE practices adoption and provided a faithful panoramic view of their organization. As for (RQ2) *Does Zeppelin help an organization envision an improvement path to follow*? All participants agreed (40% strongly agreed and 60% agreed) that Zeppelin supported them to identify areas and practices the organization succeeds and the ones that need to be improved or adopted. They also agreed that the panorama provided by Zeppelin and the practices contained in the Diagnosis Questionnaire helps define improvement actions.

When asked to provide comments and suggestions about Zeppelin, participant P3 made a comment that supports the results related to RQ2: "*This evaluation instrument can strategically support the company to understand its current state and envision its future state*". Other participants (P1 e P2) suggested to reorganize the statements contained in the Diagnosis Questionnaire to avoid repetition, since some practices of different stages are very similar (for example, there are similar statements referring to knowledge management-related practices in three stages).



Dimension	Element	Degree of Adoption				
		Stage of StH				Organization
		Agile Organization	Continuous Integration	Continuous Deployment	R&D as Innovation	
Development	Code coverage		75%			75%
	Continuous planning activities	71%				71%
	Continuous requirements engineering	50%				50%
	Focus on Feature					
	Modularized archiitecture and design		50%	75%		58%
Quality	Audits	50%	58%	75%		58%
	Automated Tests		75%	75%		75%
	Pull-Request					
	Regular Builds		75%			75%
Software Management	Agile Practice	73%	50%			71%
	Continuos delivery			75%		75%
	Continuos deployment of releases			50%		50%
	Continuos integration of work		75%			75%
Team	Contemporay and continuously evolving skills	75%			75%	75%
	Self-reflection and discipline	100%				100%
Technical Solution	Branching strategies		50%			50%
	Code review	50%				50%
	Version control		100%			100%
Knowledge	Capturing decisions and rationale	50%		75%		63%
	Continuos learning				58%	58%
	Sharing Knowledge	50%	50%	38%		44%
Operation	Logging and monitoring				100%	100%
	Reusable infrastructure				75%	75%
Business	Appropriate product ideia				100%	67%
	Management commitement	100%		75%	75%	79%
User/Customer	Involved users other stakeholders					
	Learning from usage data and feedback				70%	70%
	Proactive customers			94%		94%
	Degree of Adoption	69%	64%	70%	76%	69%

Figure 6: Adoption of CSE practices per element at each StH stage and adoption level in Org4.

The authors asked the participants about dimensions they consider most important to improve in their organizations. We noticed that the answers were aligned to data provided in the Analytics Report. Some of the aspects cited by the participants were Knowledge, Software Management, Quality, User/Costumer, Team, Operation and Business. In general, the participants found important to improve CSE practices that promote: (i) knowledge sharing and decision rationale capturing at different stages (Knowledge); (ii) adoption of good management practices to improve agile development, continuous integration and deployment (Software Management); (iii) improvement and automation in tests in different stages of software development (Quality); involvement of user and other stakeholders in all development process and learning from user data and feedback about the products (User/Costumer); (iv) self-organized, motivated and productive team (Team); (v) logging and monitoring production activities, and finally, (vi) alignment between software development, operation and business.

We observed that some dimensions less addressed by the organizations were not cited by the participants, even been pointed out in the Analytics Report. This, in fact, was expected, because Zeppelin gives a broad view of the organization, indicating aspects more and less addressed. However, which aspects will be target of improvement actions and how much they need to be improved is a decision to be made by the organization, based on its goals, needs, characteristics, constraints, etc.

Summing up, the results obtained in the study indicate that Zeppelin provided a panoramic view of the CSE practices adopted in the organization, from different perspectives, enabling to understand the organization current state and to plan improvements.

4.4 Threats to validity to study results

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 [19].

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 Diagnosis Questionnaire, which can be understood in different ways by different participants. To minimize this threat, the authors performed interviews with the participants to validate the answers. This gave us an opportunity to resolve misunderstandings. Another threat refers to the weights assigned by the authors to the adoption levels (e.g., a practice adopted at project level has weight 0.5 while an institutionalized practice has weight 1.0). This directly impacts the degree of adoption calculation. If different weights are used, the quantitative results presented in the Analytics Report may be a little different. CSE practices defined in the Diagnosis Questionnaire is also a threat. Some practices may have not been properly covered by it. To minimize this threat, CSE practices were defined based on four works addressing CSE processes and practices ([3] [8] [12][18]).

Concerning *Internal Validity*, which is concerned with the relationship between results and the applied treatment. The main threat is related to the researchers who conducted study. Two of the authors conducted the interviews to validate data. Moreover, the Analytics Reports were also elaborated by the authors. Different results could be obtained if Zeppelin had been used by the participants without the authors interference. To minimize this threat, the authors interfered as little as possible and did not influence the participants' feedback. Another threat refers to the participants providing answers not consistent with the organization reality (e.g., they could indicate a higher level of adoption for a practice than its actual level). We minimized this threat by performing interviews to validate the answers. In the interviews, we asked the participants to explain how each practice is performed in the organization, so that we could verify if the indicated level of adoption was correct.

As for *External Validity*, which is concerned with to what extent it is possible to generalize results, the main threats in this study are: (i) researchers participation; (ii) small number of organizations; and (iii) feedback obtained from only one person of each organization. Concerning (i), as discussed in the context of internal validity, the researchers participation may have influenced results. As for (ii), only five organizations were involved in the study, all of them are from the same country and most of them are small and founded in the last years. Regarding (iii), the results are based on the participants' feedback and, thus, are biased and subjective [4]. Thus, it is not possible to generalize results for cases different from the ones considered in the study. Finally, with respect to *Reliability Validity*, which refers to what extent data and analysis depend on 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 one.

In summary, considering all mentioned threats, we can only present some insights regarding Zeppelin use and generalization is limited. Thus, obtained results cannot be considered conclusive, but preliminary evidence of Zeppelin feasibility and usefulness.

5 CONCLUSIONS AND FUTURE WORKS

This paper presented Zeppelin, a diagnosis instrument that supports identify CSE practices adopted in an organization and represent them in the CSE evolutionary path. Zeppelin provides a panoramic view of CSE practices in the organization, from different perspectives. It was developed based on relevant CSE frameworks, namely StH [18], Eye of CSE [12], Continuous* [8] and FCSE [3].

Zeppelin was applied in five Brazilian software organizations: two startups, one software house, one fintech and one public organization. The results indicate that Zeppelin provides a panoramic overview which describes the current state of the adopted CSE practices in an organization, supporting to identify weakness and strengths as well as aiding in decision making about which aspects should be addressed in improvement actions.

As future works, we plan to perform new studies using Zeppelin. Moreover, we intend to provide details about the CSE practices present in the Diagnosis Questionnaire, to better support organizations to apply Zeppelin without the authors intervention. This also involves making improvements to automatically generate the Analytics Report. In addition, we intend to extend Zeppelin to provide a view from the processes perspective, considering processes present in Continuous* [8] and FCSE [3]. Finally, based on results of Zeppelin use, we intend to understand how organizations have implemented CSE practices and evolved in the CSE evolutionary path defined by the StH model [18].

REFERENCES

[1] Ian Allison. 2005. Towards an agile approach to software process improvement: addressing the changing needs of software products. *Communications of the IIMA* 5, 1 (2005), 67–76.

[2] William Sims Bainbridge. 2011. *Leadership in science and technology: A reference handbook.* Sage Publications.

[3] 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)*. ACM, 626–631.

[4] Richard L Baskerville. 1997. Distinguishing action research from participative case studies. *Journal of systems and information technology* (1997), 24–43.

[5] CMMI. 2018. CMMI for Development. CMMI.

[6] Tayana Conte, Nelson Franco, Rafael Prikladnicki, Ana Regina Rocha, Gleison Santos, Guilherme Travassos, and Kival Weber. 2015. Towards successful software process improvement initiatives: Experiences from the battlefield. (2015).

[7] Marcelo Pereira da Silva and Jacques Duílio Brancher. 2017. Sarasvati: Diagnostic method for software process improvement. In *2017 12th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE, 1–6.

[8] Brian Fitzgerald and Klaas-Jan Stol. 2017. Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software* 123 (2017), 176–189.

[9] Ivan Garcia and Luis E. Suarez. 2007. Determining Practice Achievement in Project Management using a Two-Phase Questionnaire on Small and Medium Enterprises. In *International Conference on Software Engineering Advances (ICSEA 2007)*. 75–75.

[10] Alan R Hevner. 2007. A three cycle view of design science research. *Scandinavian journal of information systems* 19, 2 (2007), 4.

[11] ISO/IEC. 2004. ISO/IEC 15504-3 Information technology — Process assessment — Part 3: Guidance on performing an assessment. ISO/IEC.

[12] Jan Ole Johanssen, Anja Kleebaum, Barbara Paech, and Bernd Bruegge. 2019. Continuous software engineering and its support by usage and decision knowledge: An interview study with practitioners. *Journal of Software: Evolution and Process* 31, 5 (2019), 21–69.

[13] Paulo S. Santos Júnior, Monalessa P. Barcellos, and Fabiano B. Ruy. 2021. *Zeppelin: A Diagnosis Instrument of Continuous Software Engineering Practices Adoption.* https://nemo.inf.ufes.br/en/projetos/continuous-software-engineering/

[14] Teemu Karvonen and et. al. 2015. Hitting the target: practices for moving toward innovation experiment systems. In *International Conference of Software Business.* Springer, 117–131.

[15] 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.

[16] Teemu Karvonen, Pilar Rodriguez, Pasi Kuvaja, Kirsi Mikkonen, and Markku Oivo. 2012. Adapting the Lean Enterprise Self-Assessment Tool for the Software Development Domain. In *2012 38th Euromicro Conference on Software Engineering and Advanced Applications.* IEEE, 266–273.

[17] Mahmood Niazi and Muhammad Ali Babar. 2009. Identifying high perceived value practices of CMMI level 2: An empirical study. *Information and Software Technology* 51, 8 (2009), 1231–1243.

[18] Helena Holmström 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 *2012 38th Euromicro Conference on Software Engineering and Advanced Applications.* 392–399.

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

[20] 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.

[21] Kerstin V Siakas, Elli Georgiadou, and Eleni Berki. 2005. Agile methodologies and software process improvement. In *IADIS International Virtual Multi Conference on Computer Science and Information Systems*, Vol. 26.

[22] SOFTEX. 2020. Reference Model for Brazilian Software Process Improvement – General Guidelines - Portuguese only. SOFTEX.