INVESTIGACIÓN
E INNOVACIÓN
EN INGENIERÍAS

Adapting DevOps to ISO 29110 through agile methodologies in Colombian VSE software developers

Adaptando DevOps a la norma ISO 29110 a través de metodologías agiles en VSE desarrolladoras de software colombianas

Manuel Alejandro Pastrana Pardo
Institución Universitaria Antonio José Camacho, Colombia
Hugo Armando Ordoñez Erazo, Carlos Alberto Cobos Lozada
Universidad del Cauca, Colombia

Recibido:
16/02/2024
Aceptado:
29/04/2024
Publicado:
21/06/2024

Correspondencia:
mapastrana@admon.uniajc.edu.co

DOI:
https://doi.org/10.17081/invinno.12.1.6916

Abstract
Objective: Determine how a VSE can implement ISO/IEC 29110 using SCRUM, XP, and DEVOPS to improve the quality of its products during the software development process. **Methodology:** 1. Integration of SCRUM, XP, and DevOps with the ISO/IEC 29110 standard, 2. Inclusion of practices in the areas recommended by ISO/IEC 29110, 3. Application of improvements and comparison between initial and final state. **Results:** The proposal is taken to seven companies, where it is initially evaluated for each area and qualification category of ISO/IEC 29110, followed by an analysis of adoption and impact to reclassify them. **Conclusions:** The implementation of agile framework practices increases the competitiveness factor of companies, especially in VSEs, providing ways to optimize their development process so that it is less empirical, more organized, and capable of implementing better quality controls that guarantee better results.

Keywords: ISO/IEC 29110 standard, DevOps, SCRUM, Very Small Entities, VSE, Software development practices

Resumen
Objetivo: Determinar cómo una VSE puede implementar ISO/IEC 29110 utilizando SCRUM, XP y DEVOPS para mejorar la calidad de sus productos durante el proceso de desarrollo de software. **Metodología:** 1. Integración de SCRUM, XP y DevOps con ISO/IEC 29110, 2. Inclusión de prácticas en las áreas recomendadas por ISO/IEC 29110, 3. Aplicación de Mejoras y comparación entre el estado inicial y final. **Resultados:** La propuesta es llevada a siete empresas, donde se evalúa inicialmente para cada área y categoría de calificación de la ISO/IEC 29110, seguida de un análisis de adopción e impacto para reclasificarlas. **Conclusiones:** La implementación de las prácticas de los marcos ágiles aumentan el factor de competitividad de las empresas, especialmente en las VSE, brindando formas de optimizar su proceso de desarrollo para que sea menos empírico, más organizado y capaz de implementar mejores controles de calidad que garanticen mejores resultados.

Palabras claves: ISO/IEC 29110 estándar, DevOps, SCRUM, Empresas Muy Pequeñas, VSE, Prácticas para Desarrollo de Software

Como citar (IEEE): M.A. Pastrana Pardo, H.A. Ordoñez Erazo, C.A. Cobos Lozada, “Adapting DevOps to ISO 29110 through agile methodologies in Co-lombian VSE software developers”, Investigación e Innovación en Ingenierías vol. 12, no. 1, pp. 189– 203, 2024, doi: https://doi.org/10.17081/invinno.12.1.6916

Introduction

The competition for the global market of software providers is currently led by multinationals, which to be competitive must ensure the quality of their products. Due to the large size of the world market, these multinationals require decentralized operations abroad to reduce costs and access skilled labor [1], so they mostly decide to outsource services to smaller companies such as VSE (Very Small Entities), as long as the reduction in operating costs does not impair the quality, process safety and reliability of the final product [2].

The VSEs, companies with less than 25 collaborators, to access the outsourcing of software services must ensure a high degree of maturity and quality in their processes. For this, in [3] they make use of software process standards such as ISO/IEC 12207, and ISO/IEC 15504 or robust models such as CMMI [4, 5], which provide a set of practices and guidelines to improve the quality in the development [6]. However, the perception of small businesses as regards the adoption and implementation of these standards is negative [7]. On one hand, these standards generate a lot of documentation and bureaucracy, which is not favorable due to the increase in the operating costs of the VSE. On the other hand, some practices do not apply to small projects, which are the most common for these types of companies [8, 9].

Bearing in mind that, worldwide, VSE software products are very important for the economy, ISO and the Joint Technical Committee for IT Standards (JTC1), defined ISO/IEC 29110 (Hereinafter ISO/IEC 29110, as it is normally known) to provide standardization mechanisms of practices in the improvement of software development processes, which allow them to be included and competitive in the global market [10]. ISO/IEC 29110 defines the set of practices to be followed and the minimum documents necessary to have a development process based on the quality of its deliverables [1].

Practice in ISO/IEC 29110 is organized into nine categories according to [11], namely: 1) Project management, 2) Requirements analysis 3) Functional and physical architecture, 4) Construction and unit testing, 5) Version control, 6) Integration and testing, 7) Verification and validation, 8) Product delivery, and 9) Self-assessment.

The implementation of these practices in VSEs has been based on agile frameworks such as SCRUM and XP. This is because customers (the multinationals) are increasingly demanding, have higher expectations regarding the content of applications, demand an increase in software quality, and require shorter delivery times [12]. In this scenario, the responses must be faster and faster, with greater stability, safety, and predictability. That is, a mix of stability and reliability, it is here that DevOps takes a leading role, fostering an environment of organizational culture based on collaboration and communication between the Development (Dev) and Operations (Ops) teams in the development companies of software [13,14].

DevOps aims to reduce the time that elapses from the moment a change is uploaded to the version control repository of the software being built (commit), until that change is put into production, with no reprocessing while ensuring high quality in the software. In this sense, this research articulates the practices defined by ISO/IEC 29110 with SCRUM, XP, and DevOps, for which the members of the VSE adopt a culture that has the main characteristic of ensuring automation and monitoring in all the steps of construction of the software, from integration, testing, and release to implementation and management of infrastructure [11].

Based on the foregoing, this work features the following contributions: 1) Implementation of ISO/IEC 29110 practices complemented with those of DEVOPS, for a set of 7 VSE software developers of southwestern Colombia; 2) The mapping of the ISO/IEC 29110 practices that can be articulated with DEVOPS; 3) The definition of each of the DEVOPS environments and the tools that can be used according to each practice of ISO/IEC 29110; and 4) The appropriation of business regulations for the management of configuration and deployment.

The content of the rest of the article is organized as follows: Section 2 presents the previous work related to this research; Section 3 describes the way ISO/IEC 29110 was integrated with DevOps in VSE and describes the evaluation of the implementation of the practices defined in the seven VSE software developers; and Section 4 presents conclusions and future work the research group expects to carry out in the short term.

Related Works

Adopting and implementing changes in the processes and procedures traditionally carried out within companies is always complex and usually requires a significant investment in time and effort. As such, every time an organization adopts a technology, methodology, or work approach, such adoption must be driven by a business need. In software development companies, the adoption of practices based on ISO/IEC 29110 and DEVOPS is a vital necessity, which seeks to increase the competitiveness of the organization. Based on this, the works presented below highlight the fundamental practices for both ISO/IEC 29110 and the implementation of DEVOPS.

About DevOps

In 2016, DevOps was defined as an emerging paradigm that aimed to integrate developers closely with operations personnel [15]. They established that it is a critical competitive advantage to be able to respond quickly to cultural and organizational changes and to constantly promote new approaches, tools, and open-source artifacts to implement applications in the cloud through an automated process. For this, they proposed a systematic classification of DEVOPS artifacts to transform them into TOSCA (Topology and Orchestration Specification for Cloud Applications), an emerging standard that proposes mechanisms to allow portability between different clouds. The proposal was implemented through a chain of open-source tools from end to end. In addition, the practical feasibility of the proposed approach was validated with a case study.

In 2017, it was determined that organizations are introducing agile software development techniques in their operations to increase the pace of their development process, and to improve its quality. In [16], a systematic review of the literature was presented highlighting what researchers have written about DEVOPS. They go on to describe the results of an exploratory study based on interviews involving six organizations of various sizes that are active in the implementation of DEVOPS. As findings, all organizations were seen to enjoy positive experiences, and adopting DEVOPS brought only minor problems.

In 2017, DEVOPS was defined as a key mechanism for the frequent and reliable updating of a system in its operational state. To do this, it is based on collaboration and multifunctional automation between software development and operations, focused on the changes required in technical, organizational, and cultural aspects. In [17], an exploratory study was presented of how DEVOPS is implemented in the development of web applications and services in small and medium enterprises. Five different development contexts were studied with DEVOPS implementations, in which benefits such as quick releases and minimum implementation errors were achieved. The data was mainly collected through interviews with 26 professionals and observations made in the companies.

In 2018, regulatory authorities were deemed to require critical testing of the software before its acceptance on the market. To meet this challenging scenario, the use of DEVOPS was recommended, in which integration and continuous implementation are the default practices [18], but do not match the regulatory standards of software development. To address this fact [19], the objective was to approach DEVOPS and the development of regulated software. For this, 1) it was proposed that developers use tools and practices with which they are familiar; and

2), regulatory authorities are allowed to create confidence in the solutions provided by manufacturers by defining a mapping between DEVOPS and regulated software development. As a result, a close integration was obtained between development tools, requirement management, version control, and deployment channeling, thus creating development practices that comply with regulations. With the above, an incremental improvement in the approval of software quality was demonstrated before the actual implementation.

In 2019, the authors of [12] detail 15 real scenarios of companies from different domains that successfully adopted DevOps. For this adoption, a model (workflow) was defined, which was evaluated through a case study in an institution of the Brazilian government, through a focus group that allowed it to collect the company's perceptions of the adoption of DEVOPS. The case study details the real scenarios and explains the role of each ISO/IEC 29110 category during the adoption of DEVOPS, providing evidence that collaboration is the main factor in DEVOPS, and not just automation and tools.

In 2019, DevOps was also defined as a set of software engineering practices that aims to have shorter software development cycles while the high degree quality of the system being built is still [20]. This is achieved using a highly automated chain of tools. In [21] DEVOPS is used for the development of a modular system of control and acquisition of data based on components for a device called the Wendelstein 7 X (W7 X) stellarator, a fusion device composed of complex systems that require high reliability, availability, and quality of all participating components. In this scenario, in the construction of the components of the system, new functions must be developed, errors corrected, or performance problems improved in a very short time. In addition, they must implement new software packages, establish new configurations, change rules and restrictions, and improve experiment planning views. For all this to work harmoniously, a continuous integration and deployment environment is necessary, typical of DEVOPS practices.

About ISO/IEC 29110

In 2010 [10], the International Organization for Standardization (ISO) defined ISO/IEC 29110 to help increase competitiveness in small software development organizations. Due to its recent appearance, implementing the standard is still difficult for small organizations, because of poor knowledge and the costs associated with implementation [22]. As such, in 2014 [23] to reduce the conceptual gap, a process model (UP VSE) based on the Unified Process was proposed, which implements the requirements engineering practices of ISO/IEC 29110-5-1-1 for companies to have an illustrative and reusable case that allows them to accelerate the interpretation and implementation of the standard. The proposal was empirically evaluated in the definition of the PRODIGIA development process in the context of a research and development group in industrial automation.

In 2016 [1], the details of a case study involving nine VSEs were described, in which a pilot was conducted of the new profiles and ISO/IEC 29110 guides for the software development life cycle for very small entities. The pilot was specifically designed to address the standardization needs of these entities. The purpose of this work was to disseminate the first success stories in the pilot trials of this standard. The lessons learned from these case studies assist in the adoption of this standard in an industrial environment so that VSEs can enter the global software development market.

For [2] in 2017 and [24] in 2018, most software development organizations worldwide are very small entities (VSE) that do not have the experience to adopt or adapt software engineering standards to meet their needs. Therefore, several countries that had not participated in the development of international standards decided to join the ISO working group with the mandate to develop the ISO/IEC 29110 series. The guides developed are easily understandable and freely available, which has greatly helped its adoption. More than 17 countries, such as Colombia, Brazil, Haiti, Jordan, Malaysia, Mexico, Peru, and Thailand are teaching ISO/IEC 29110. Many VSEs are using ISO/IEC 29110 to develop their first products. Many countries have adopted ISO/IEC 29110 as a national standard and have advanced low-cost independent government certification and evaluation programs that allow VSEs to demonstrate recognition of their competencies to local and international clients and partners.

In 2018 [8], a systemic study was presented which involves around 76% of the companies that develop software in Mexico, characterizing them as small and very small entities. It emphasizes that the ISO/IEC 29110 standard was designed for this type of company. However, the lack of a culture of processes, and guides and the lack of support tools makes it difficult to implement them within organizations of this type. The study presents a review of the literature to identify how the implementation of this standard is being carried out, focusing on the use of tools and practices from other models, standards, or agile frameworks that support its implementation.

The work of [9] in 2019 concentrated on the use of the standard to improve the life cycle of small software development entities, which have recently begun to be implemented worldwide. To do this, it uses ArchiMate, a widely accepted open standard for modeling Enterprise Architecture backed by a variety of modeling tools, which is based on 1) the development of a basic profile metamodel for ISO/IEC 29110; 2) the mapping definition of the basic profile metamodel for ISO/IEC 29110 with ArchiMate; 3) its evaluation using the Bunge Wand Weber (BWW) model; and 4) the mapping application that results in the development of the Basic Profile ArchiMate Model for ISO/IEC 29110 that is implemented in a modeling tool that is freely available and can be used by VSE.

From the study of related works [25], most of the research related to DevOps could be identified as focused on tools, integration of collaboration, development practices, and continuous deployment. Although practices and tools are important in the DEVOPS environment, none of the studies was interested in defining the guidelines that must be followed before implementing DEVOPS in a VSE. These make it possible to identify the current state of the practices and which practices would be easiest to implement in the context of the company. Moreover, the studies based on the ISO/IEC 29110 standard focus on the adoption of the standard itself, or the implementation of practices in the development process, process indicators, certifications, and as such compliance with it. While this is good, it is possible to potentiate good development practices in VSE, integrating DevOps to good development practices of ISO/IEC 29110, to have better management in version control, increase the number of successful deployments, do code tests, and integrate functional tests before a VSE delivers software to production.

Integration of DevOps with the ISO/IEC 29110 standard -Discussion and results

DEVOPS consists of a set of practices implemented through different techniques and tools that, although focused on development, are not only limited to this [12]. Its main objective is to become part of the organizational culture to maintain quality as the central axis of the development process, largely minimizing reprocesses [13]. This way of shaping the organizational culture is consistent with XP and SCRUM, which focus their roles, events, and artifacts in a continuous interaction, allowing the product to evolve from small deliveries to a final product with the maximum possible quality in each delivery as ISO 29110 do too [26]. For this reason, is necessary to map which practices are commonly used in VSEs between ISO29110, DEVOPS, SCRUM, and XP. In this regard, Table 1 shows the detail below.

Table 1: Comparison of practices made by authors between ISO/IEC 29110, SCRUM, XP, and DEVOPS.

ISO 29110 categories	SCRUM	XP	DEVOPS
Project management	Practices: • Estimation by SCRUM. • Prioritization by MoSCoW • SCRUM board Events: • Daily SCRUM		
Requirement analysis	Artifacts: • Product backlog with user histories • BP models		
Physical and functional architecture	Artifacts: • Software Architecture Document (SAD)	Practice: • Define design standard	Practice: • Standardization
Construction and unit tests	Practice: • Incremental iterative development by sprints.	Practice: • Collective ownership of the code	Practices: • Programming standard • Unit tests • Static code analysis
Version control	Event: • Daily SCRUM	Practice: • Collective ownership of the code	Practice: • Implementation of versioning system
Integration and testing	Practice: • DOD (Definition Of Done)	Practice: • Continuous integration • Unit tests	Practice: • Continuous integration • Unit tests
Verification and assessment	Event: • Daily SCRUM Practice: • SCRUM board	Practice: • Sustainable rhythm	Practice: • Statistical analysis of code
Product delivery	Event: • Sprint review	Practice: • Continuous deployment	Practice: • Continuous deployment
Self-assessment	Event: • Sprint retrospective		Practice: • Retrospective reviews.

Source: Own Elaboration.

Additionally, is important to highlight that for companies, adopting agile frameworks in their entirety is not completely possible, due to the needs of their customers, who constantly demand certain things from traditional models. For example, planning commitments measured in hours, non-rapid adaptation to changes, and the development of test cycles at the end of the entire development. In [14] it is suggested that it is possible to achieve compliance with sprints effectively, generating an awareness of the development process more assertively and including in the development stage the practices suggested by XP, but materialized in tools that allow complying with the provisions of DEVOPS. Therefore, each of the categories of the ISO/IEC 29110 standard must identify the set of practices, tools, and artifacts to be carried out that allow defining a development process focused on preventive quality and according to the DevOps model.

Project Management of Metadata

Using user stories grouped in the product backlog, it is possible to have the information required to start the project more quickly and with the direct participation of the user [11], all in a constant evolution and refinement that allows reaching a degree of adaptation to change more in line with SCRUM. Thanks to this, the SCRUM board can be implemented in each project according to [27]. This board allows for properly managing the project by identifying the daily progress, delays caused by impediments, the tasks completed according to the Definition Of Done (DOD) described by [27], and even the overload of the team in development activities. The DOD implies that there are three minimum states for all activity placed on the board: to do, doing, and done. Other states can be added if the team so wishes. Therefore, this practice helps the team to self-manage their progress and react quickly to risks that are about to impact or minimize those that have already affected the normal course of the project.

Another practice suggested within board management is that one person may be attending to only one activity at a time. This enables attention to be kept fully focused on that activity until the DOD is complied with, which implies that the activity can only be closed until the person carrying it out has performed their own tests that ensure a minimum degree of quality.

In [27] it is suggested that every project be constructed using an iterative and incremental development cycle, so the suggested measure to be used for the division of work to be performed is the sprint. During a sprint, development teams work for a period from one to four weeks. Within this time the effort is not measured in the number of hours it takes to do so, but in commitments made. The commitment is defined as the user stories committed to be carried out during this time.

To know in what order these user stories should be performed, a prioritization technique must be used to identify the business value contributed to the project when completing the story, as indicated in [27]. The technique suggested to companies, owing to its simplicity, is MoSCoW [27]. The capital letters each represent a phrase in English that identifies the degree of value involved in the construction of the story. Each letter of MoSCoW is described below:

- *Must have (M):* These functionalities are crucial for the end user. They are the most important for the project and must be done first.
- *Should have (S):* These functionalities represent a need that ought to be done in second place for the end user and hold back sprints that follow the first ones.
- *Could have (C):* These are non-core functionalities that can be seen as an advantage. These features are desirable, but if they are not included, the operating result is not affected.
- *Won´t have (W):* These functionalities will not be carried out, at least in the current phase of the project.

The sets of practices suggested allowing teams to maintain evolutionary control of the project and generate continuous monitoring of the information of progress that supports decision-making involving everyone, avoiding incurring delays, managing risks, and promoting the correct use of time for the fulfillment of the objectives of the sprint.

Within the framework of the case studies carried out for this investigation, both at the beginning and at the end of the process, a survey was applied, allowing identify the state of each company according to [11], The possible states are namely: Initial, The company works in a completely empirical way, there is no development standard, project management, and the stages of the development cycle are done without following any guidelines; Basic, the company has a formalism for its different stages, but they do not implement international standards accepted by the industry or best practices; Intermediate, the company knows and uses some standards and best practices; and Advanced, the company knows very well the process of internal software development due to its standardization, all its artifacts are based on accepted models and internationally recognized and the use of best practices is part of the organizational culture.

The first four companies that will be mentioned from this point were participants in the ICT Cluster project of the Colombian Pacific (PacifiTic). The remaining entities independently decided to implement the process according to the satisfactory results from PacifiTic. All of them, are software development companies that are composed of less than 20 employees. Also, they are in the city of Cali, Colombia. All their teams are composed of 1 to 5 members depending on the size of the project and the client. Often, they have from 1 to 5 projects at the time, and some of them are from support services. Table 2 below shows the contrast between before and after the implementation of the practices in the seven companies.

Table 2. Comparison of initial state vs final state for project management categories.

Entity	Initial state	Final state
VSE1	Basic	Intermediate
VSE2	Basic	Intermediate
VSE3	Intermediate	Advanced
VSE4	Initial	Basic
VSE5	Basic	Intermediate
VSE6	Basic	Intermediate
VSE7	Intermediate	Advanced

Source: Own Elaboration.

Physical and functional architecture

As a complement to requirements analysis, the identification of restrictions in the acceptance criteria of user stories was included. The user story now included nonfunctional needs. For example, how long can they wait to generate a report? (5 seconds, 10 seconds, is more time unacceptable?). These restrictions identified from the user stories allow for defining quality attributes at the design stage of the architecture. They are not functionalities of the solution but behaviors that must be met to obtain the desired operating result [29]. Some examples of this might be usability, maintainability, modifiability, scalability, performance, security, and reliability. These attributes are resolved in software development using design patterns.

Once all the companies internalized the appropriate way to include restrictions in the acceptance criteria, they needed to know how to describe the solution through diagrams using UML (Unified Modeling Language) correctly and effectively by their development teams. Therefore, in the consultancy it was necessary to conduct training in UML, highlighting how various design patterns can be used in the solution and how these are reflected in the possible architectural diagrams [29]. The suggested practice to be able to carry out this information continuously and efficiently is to standardize an artifact that allows them to have a list of quality attributes and frequent patterns in most of their projects, as a guide, and some examples of base diagrams that allow starting from that point to the layout of the current solution. This artifact is called a software architect document.

At the beginning and the end of the process, a survey was applied to identify the state of the company. Table 4 below shows the contrast before and after the implementation of practices in companies.

Table 4: Comparison of initial vs. final state for the functional and physical architecture category.

Entity	Initial state	Final state
VSE1	Basic	Intermediate
VSE2	Basic	Intermediate
VSE3	Basic	Advanced
VSE4	Basic	Intermediate
VSE5	Basic	Intermediate
VSE6	Basic	Intermediate
VSE7	Intermediate	Advanced

Source: Own elaboration.

Construction and unit tests

To understand how companies were building their applications, and to determine the good development practices that should be adopted, it was first necessary to identify which bad practices were recurring in their projects. Therefore, the following concepts were previously addressed:

- The Blob: A poor implementation of software that is never refined and persists in the software (difficult to refactor). This practice generally affects performance, maintainability, modifiability, and scalability.
- Continuous Obsolescence: Products continue to evolve in the market; however, the underlying technology (frameworks used in development, programming language version, database engines) may become obsolete and not supported.
- Lava Flow: Code whose function is not known, but which is not eliminated to prevent damage to the product in general.
- Functional Decomposition: A true object-oriented design is not made, and its programming obeys that functional organization. Classes with a single method that groups the entire solution within itself and can be broken down into several classes with defined responsibilities.
- Poltergeist: Classes that appear and disappear as ghosts. Its sole purpose is to initiate the interaction of other classes. Some examples of this can be Start_Routine and Start_Process. They are unnecessary and consume many resources.
- Golden Hammer: Over time, development teams become "Experts" in the use of a particular programming language, database, development frameworks, etc. Therefore, any new project is tackled exactly as it has always been done, without stopping to analyze the pros and cons of the proposed solution.
- Spaghetti Code: Messy code that is easier to redo than try to understand. It is usually not object-oriented.
- Walking Through a minefield: This is the result of never testing the code in a test environment, but only on the local computer. Therefore, some

problems that never appeared in the local environment may arise during implementation.
- Cut and Paste Programming: This is the result of carrying out cut and paste. If the code is not understood before being cut, all problems and errors can be expanded throughout the project.

Table 5: Comparison between initial vs. final state for the construction and unit tests category.

Entity	Initial state	Final state
VSE1	Basic	Intermediate
VSE2	Basic	Intermediate
VSE3	Intermediate	Advanced
VSE4	Intermediate	Advanced
VSE5	Initial	Intermediate
VSE6	Initial	Intermediate
VSE7	Intermediate	Advanced

Source: Own elaboration

Version control

Companies VSE1, VSE4, VSE5, VSE6, and VSE7 at the beginning of the project already used GitLab as a version control system. This version control system implements the best practices suggested by [14] to guarantee collective ownership of the code and recover quickly from errors. This is very much aligned with DEVOPS and suggested by [13] that his work details the impact of having a tool that controls changes continuously and allows the team to ensure that the source generated in the daily increase is protected against loss of information. VSE2 used SVN as a versioner, however, in explaining the possibilities that GitLab has as described by [14] they decided to move to this tool. VSE3 used TFS (Team Foundation Services), and it was not necessary to change the tool, since TFS supports all the requirements of the ISO standard.

It was recommended to all companies that the work be organized into two branches master and development. All master code must be ready to be implemented in the production environment. Only work completed and tested through Software Quality Assurance (SQA) can be in the master branch (to ensure it is stable). The second branch includes the code developed during the sprint. Every time there is a merger between the development branch and the master branch, a new software version is created. In addition, when a new version is in production, a new release version must be created (release). This set of branches allows the team to recover from a problem quickly. A best practice suggested to companies was the use of quick fixes (hotfixes) and errors (bugs). These branches are used to correct errors detected when the software is already in production. Once the correction is made, it is merged within the development branch and then within the master branch creating a new version in the launching branch.

The version tool suggested also includes a plugin called GitFlow, which creates continuous integration and deployment. This tool can be used to replace other specialized ones for these actions such as Jenkins, TravisCI, and Heroku. All companies adopted the version tool without GitFlow. However, they want to implement it in the future. The results before and after are shown in Table 6.

Table 6: Comparison between initial vs. final state for the version control category.

Entity	Initial state	Final state
VSE1	Intermediate	Advanced
VSE2	Basic	Intermediate
VSE3	Intermediate	Advanced

VSE4	Basic	Intermediate
VSE5	Initial	Intermediate
VSE6	Initial	Intermediate
VSE7	Intermediate	Advanced

Source: Own Elaboration

Verification and validation

The seven companies were aware of the importance of collaborative work and constant communication but were unaware of adequate mechanisms to implement good DEVOPS practices at this point. Based on [24], it was proposed to use the daily SCRUM, an event that makes it possible to verify the progress of the project increase and validate if there are risks that are about to affect this progress or that are already affecting it. In this meeting, the team in front of the SCRUM board identified the progress achieved through the question "What did I do yesterday?" Additionally, they verify the work that will be done during the day with the question "What will I do today?" And finally, they socialize the problems they have through the question "What problems or difficulties have I had?", all to achieve the correct fulfillment of the objectives of the sprint. This practice allows the team to be constantly informed about what is happening in the project, improve the development culture, and have a high degree of transparency about the progress.

Additionally, and for a quick validation of the quality of what is being developed, static code analysis tools are suggested. These tools allow for identifying in development time how the quality of the code is evolving concerning maintainability, modifiability, and security by measuring bad practices present in the code. The conscious use of these tools transforms the culture of development, making work cleaner and minimizing the impact of reprocessing, maintenance, and even the transfer of knowledge between team members. The before and after results are shown in Table 8.

Table 8: Comparison between initial vs. final state for the verification and validation category.

Entity	Initial state	Final state
VSE1	Basic	Intermediate
VSE2	Basic	Intermediate
VSE3	Intermediate	Advanced
VSE4	Intermediate	Advanced
VSE5	Basic	Intermediate
VSE6	Basic	Intermediate
VSE7	Intermediate	Advanced

Source: Own Elaboration

Product delivery

The Jenkins tool not only allows continuous integration. In the same way as some versioners such as those based on Git, it allows continuous deployment. This practice suggested by DEVOPS and born from XP, specifies that the faster it is ready for deployment, the better the result. Therefore, the changes generated in the versioner and integrated by the continuous integration tool are deployed in a test environment automatically for the required validations from the functional aspect. Previously, the companies having finished construction of the deliverable, generated a dropdown to be sent by email to the infrastructure area, who attended to them in the order in which they received the requests. This dropdown, without prior verification, and even occasionally without deployment instructions, was placed directly on the test server in the best cases, and there the quality and development team were notified to check that it was correct. The commercial manager of the application was then notified, so that it would be validated before notifying the client of the deployment. Any problem presented at this point returns the team to the development stage to start this process again.

Using Jenkins, a continuous delivery model was implemented and adapted to the specific needs of each company. In all cases, the final version was not delivered to the production server, but to a test server as indicated above, where the quality of that delivered is validated by acceptance tests, and it is determined if it is ready to move on to production. This strategy was adopted because most companies offer no customized services and developments. In addition, incremental functional demonstrations can be made to customers, which enables them to see the progress made during the development process. The before and after results are shown in Table 9.

Table 9: Comparison between initial vs. final state for the product delivery category

Entity	Initial state	Final state
VSE1	Basic	Intermediate
VSE2	Initial	Intermediate
VSE3	Basic	Intermediate
VSE4	Basic	Intermediate
VSE5	Basic	Intermediate
VSE6	Basic	Intermediate
VSE7	Intermediate	Advanced

Source: Own Elaboration

Self-assessment

One of the best practices produced by agile models such as SCRUM, which allows continuous improvement, is retrospectives. DevOps models, as indicated [12], seek that development teams transform their development culture, empowering it beginning with recognizing how they carry out their work, what they might improve on, and how they should improve. Thus, the various retrospective techniques allow alignment with DEVOPS to achieve this objective. None of the companies with which the consultancy was performed had used retrospective models and, therefore, was not able to learn from prior projects. As such, the implementation of this suggestion started with the completion of the project. It is important to highlight that the success of the proposed approach depends on the support of the company's managers and the dedication of the staff to these activities [15].

Table 10 reflects the comparison between before and after the implementation of the mentioned improvements.

Table 10: Comparison between initial vs. final state for self-assessment category

Entity	Initial state	Final state
VSE1	Initial	Intermediate
VSE2	Initial	Intermediate
VSE3	Initial	Intermediate
VSE4	Initial	Intermediate
VSE5	Initial	Intermediate
VSE6	Initial	Intermediate
VSE7	Initial	Intermediate

Source: Own Elaboration

Conclusion

In this work, a viable and proven way was presented by seven VSE software developers in the southwest of Colombia, to implement DevOps hand in hand with the ISO/IEC 29110 standard. To do this, the best practices of the most used agile methodologies were first mapped in these types of organizations and those that required to be implemented. Additionally, the mapping made it possible to identify in which phases and what practices can be potentialized by integrating the DevOps culture into the daily tasks of the VSE. Once the practices carried out by the VSEs were incorporated, the changes in the development processes were notable. In general, all the VSEs managed to better understand the estimate, since more activities related to internal quality, meetings, and formal deliveries to the customer were considered.

Similarly, DEVOPS practices allowed the refinement of version control, the deployment, and the refinement of code along with continuous integration. It is important to highlight that the above goes hand in hand with the generation of strategies to reduce "technical debt" measures with tools like SonarQube through continuous improvement.

As an important aspect, the commitment to quality in each of the VSEs was identified, since the best practices of DEVOPS, as of the ISO/IEC 29110 standard, became organizational policies to be applied in all the projects that are developed within each one of the VSE research participants.

.Future Works

In the future, it is expected to define a maturity model that allows companies to have a guide for the implementation of best practices fully adapted to the context of VSE, based on DEVOPS and ISO/IEC 29110, this work being a high precedent impact for the region. To achieve this, it is necessary to define a set of metrics aligned with the most common strategic objectives of the VSE, which can be traced by the managers of the companies, on the progress of the implementation and the impact, on time and money, of the projects where the initial pilot is carried out. Additionally, to identify which are the most difficult practices to implement and refine the implementation strategies that facilitate better results.

Acknowledgments

The authors express their gratitude to the Universidad del Cauca and the Institución Universitaria Antonio José Camacho for participating in the PhD research project entitled "Guide for the implementation of harmonized software development practices based on SCRUM and DEVOPS in very small companies" and supporting provided to the group of researchers in the execution of this project.

Bibliographic References

[1] C. Y. Laporte and R. V. O'Connor, "Implementing process improvement in very small enterprises with ISO/IEC 29110: A multiple case study analysis," in Proceedings - 2016 10th International Conference on the Quality of Information and Communications Technology, QUATIC 2016, 2017, pp. 125–130. DOI: 10.1109/QUATIC.2016.033.

[2] C. Y. Laporte, M. Muñoz, and B. Gerançon, "The education of students about ISO/IEC 29110 software engineering standards and their implementations in very small entities," IHTC 2017 - IEEE Canada International Humanitarian Technology Conference 2017, pp. 94–98, 2017, DOI: 10.1109/IHTC.2017.8058208.

[3] M. Chaabane, I. Bouassida, R. Colomo-palacios, W. Gaaloul, and M. Jmaiel, "Science of Computer Programming A modeling approach for Systems-of-Systems by adapting ISO / IEC / IEEE 42010 Standard evaluated by," Sci Comput Program, vol. 184, p. 102305, 2019, DOI: 10.1016/j.scico.2019.102305.

[4] I. Keshta, "A model for defining project lifecycle phases: Implementation of CMMI level 2 specific practice," Journal of King Saud University - Computer and Information Sciences, 2019, DOI: 10.1016/j.jksuci.2019.10.013.

[5] D. Proença and J. Borbinha, "Formalizing ISO/IEC 15504-5 and SEI CMMI v1.3 – Enabling automatic inference of maturity and capability levels," Comput Stand Interfaces, vol. 60, no. April, pp. 13–25, 2018, DOI: 10.1016/j.csi.2018.04.007.

[6] A. D. Robinson, "Very small entities (VSE); The final systems engineering (SE) frontier," in 12th Annual IEEE International Systems Conference, SysCon 2018 - Proceedings, 2018, pp. 1–4. DOI: 10.1109/SYSCON.2018.8369570.

[7] M.-L. Sánchez-Gordón and R. V. O'Connor, "Understanding the gap between software process practices and actual practice in very small companies," Software Quality Journal, vol. 24, no. 3, pp. 549–570, Sep. 2016, DOI: 10.1007/s11219-015-9282-6.

[8] M. Munoz, J. Mejia, and A. Lagunas, "Implementation of the ISO/IEC 29110 standard in agile environments: A systematic literature review," in Iberian Conference on Information Systems and Technologies, CISTI, 2018, pp. 1–6. DOI: 10.23919/CISTI.2018.8399332.

[9] A. Buchalcevova, "Using ArchiMate to model ISO/IEC 29110 standard for very small entities," Comput Stand Interfaces, vol. 65, no. February, pp. 103–121, 2019, DOI: 10.1016/j.csi.2019.03.001.

[10] T. Varkoi, "Process assessment in very small entities - An ISO/IEC 29110 based method," in Proceedings - 7th International Conference on the Quality of Information and Communications Technology, QUATIC 2010, 2010, pp. 436–440. DOI: 10.1109/QUATIC.2010.80.

[11] H. Ordoñez, M. Pastrana, C. Castiblanco, O. Rincón, and A. Ordonez, "A model approach to ISO-29110 for very small enterprises," in XXII Ibero-American Conference on Software Engineering, CIbSE 2019, 2019, pp. 517–530.

[12] W. P. Luz, G. Pinto, and R. Bonifácio, "Adopting DevOps in the real world: A theory, a model, and a case study," Journal of Systems and Software, vol. 157, p. 110384, 2019, DOI: 10.1016/j.jss.2019.07.083.

[13] L. E. Lwakatare et al., "DevOps in practice: A multiple case study of five companies," Inf Softw Technol, vol. 114, no. June, pp. 217–230, 2019, DOI: 10.1016/j.infsof.2019.06.010.

[14] M. Pastrana, H. Ordóñez, A. Rojas, and A. Ordoñez, "Ensuring Compliance with Sprint Requirements in SCRUM," in Advances in Intelligent Systems and Computing, 2019, pp. 33–45. DOI: 10.1007/978-981-13-6861-5_3.

[15] J. Wettinger, U. Breitenbücher, O. Kopp, and F. Leymann, "Streamlining DevOps automation for Cloud applications using TOSCA as

standardized metamodel," Future Generation Computer Systems, vol. 56, pp. 317–332, 2016, DOI: 10.1016/j.future.2015.07.017.

[16] F. M. A. Erich, C. Amrit, and M. Daneva, "A qualitative study of DevOps usage in practice," in Journal of Software: Evolution and Process, 2017. DOI: 10.1002/smr.1885.

[17] A. Balalaie, A. Heydarnoori, and P. Jamshidi, "Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture," IEEE Software. 2016. DOI: 10.1109/MS.2016.64.

[18] N. Forsgren and M. Kersten, "DevOps metrics," Commun ACM, vol. 61, no. 4, pp. 44–48, 2018, DOI: 10.1145/3159169.

[19] T. Laukkarinen, K. Kuusinen, and T. Mikkonen, "Regulated software meets DevOps," Inf Softw Technol, vol. 97, no. December 2017, pp. 176–178, 2018, DOI: 10.1016/j.infsof.2018.01.011.

[20] G. Rong, Z. Jin, H. Zhang, Y. Zhang, W. Ye, and D. Shao, "DevDocOps: Towards Automated Documentation for DevOps," in Proceedings - 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice, ICSE-SEIP 2019, in ICSE-SEIP '19. Piscataway, NJ, USA: IEEE Press, 2019, pp. 243–252. DOI: 10.1109/ICSE-SEIP.2019.00034.

[21] M. Lewerentz et al., "Implementing DevOps practices at the control and data acquisition system of an experimental fusion device," Fusion Engineering and Design, vol. 146, no. October 2018, pp. 40–45, 2019, DOI: 10.1016/j.fusengdes.2018.11.022.

[22] W. Lopez, W. Bernuy, and B. Pando, "Bringing a Micro Enterprise towards the ISO 29110," in 2019 14th Iberian Conference on Information Systems and Technologies (CISTI), IEEE, Jun. 2019, pp. 1–6. DOI: 10.23919/CISTI.2019.8760865.

[23] J. J. Alvarez and J. A. Hurtado, "Implementing the software requirements engineering practices of the ISO 29110-5-1-1 standard with the unified process," in 2014 9th Computing Colombian Conference, 9CCC 2014, 2014, pp. 175–183. DOI: 10.1109/ColumbianCC.2014.6955337.

[24] A. D. Robinson, "Very small entities (VSE); The final systems engineering (SE) frontier," in 12th Annual IEEE International Systems Conference, SysCon 2018 - Proceedings, 2018, pp. 1–4. DOI: 10.1109/SYSCON.2018.8369570.

[25] J. D. Millán, J. N. Cudina, and J. C. Ossa, "Criticism and Psychology: The Understanding with a Postmodern Psychology," Human Arenas, vol. 3, no. 3, pp. 343–359, 2020, DOI: 10.1007/s42087-019-00083-9.

[26] M. Muñoz and M. Negrete, "Reinforcing DevOps Generic Process with a Guidance Based on the Basic Profile of ISO/IEC 29110," in Advances in Intelligent Systems and Computing, Springer, Ed., Springer.Springer Cham, 2020, pp. 65–79. DOI: 10.1007/978-3-030-33547-2_6.

[27] K. Schwaber and J. Sutherland, "The Scrum Guide: The Definitive the Rules of the Game," Scrum.Org and ScrumInc, 2017, DOI: 10.1053/j.jrn.2009.08.012.

[28] M. Pastrana, H. Ordóñez, A. Ordonez, L. H. Thom, and L. Merchan, "Optimization of the Inception Deck Technique for Eliciting Requirements in SCRUM Through Business Process Models," Business Process

Management Workshops, vol. 4928, no. January, pp. 649–655, 2018, DOI: 10.1007/978-3-319-74030-0_52.

[29] R. S. Pressman and B. R. Maxim, Software Engineering: A Practitioner's Approach, Eighth Edition, McGraw-Hil. Boston, USA: McGraw-Hill, 2015.

[30] M. Cohn, User Stories Applied: For Agile Software Development, vol. 1, no. 0. 2004. DOI: 10.1017/CBO9781107415324.004.

[31] H. Ordóñez, A. F. E. Villada, D. L. V. Vanegas, C. Cobos, A. Ordóñez, and R. Segovia, "An impact study of business process models for requirements elicitation in XP," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2015, pp. 298–312. DOI: 10.1007/978-3-319-21404-7_22.

[32] J. Martínez Garcés y J. Barreto Fereira, "Modelo de planeación para la inversión tecnológica en centros de investigación universitarios", Investigación e Innovación en Ingenierías, vol. 7, n.º 2, jul. 2019. DOI: https://doi.org/10.17081/invinno.7.2.3448

[33] J.M. Munive - Álvarez, "Calidad de la iluminación en las aulas de clase en una Institución de Educación Superior", Investigación e Innovación en Ingenierías, vol. 8, n°. 1, 2020. DOI: https://doi.org/10.17081/invinno.8.1.3409

[34] M. Pastrana, H. Ordóñez, A. Ordóñez, and L. Merchan, "Requirements elicitation based on inception deck and business processes models in scrum," in Communications in Computer and Information Science, 2017, pp. 327–339. DOI: 10.1007/978-3-319-66562-7_24.