Evolution of DevOps: Lessons learned for success as part of digital strategy

Evolución de DevOps: Lecciones aprendidas para el éxito como parte de la estrategia digital

Gabriel Silva-Atencio
Universidad Latinoamérica de Ciencia y Tecnología (ULACIT), Costa Rica
gsilvaa468@ulacit.ed.cr
https://orcid.org/0000-0002-4881-181X
Mauricio Umaña-Ramírez
Universidad Católica de El Salvador (UNICAES), El Salvador
mauricio.umana@catolica.edu.sv
https://orcid.org/0000-0002-0733-5183

Recepción: 08 Junio 2023
Aprobación: 18 Octubre 2023

Acceso abierto diamante

Abstract

The objective was to investigate the good practices that must be followed to implement DevOps culture to accelerate the development of software applications in shorter periods and unify the story so they can work hand in hand. The research was under a qualitative approach and a review of references to provide information on the terms and development methodologies used. Finally, to put DevOps into practice, it is necessary to understand that, even if you have the best tools available, if you do not have the culture, you will not get better results since it is a combination of good practices to be able to function. In this sense, to collaborate with this culture change, the support of trained leaders is essential, who support and sponsor the new way of working, where all areas of the company can be integrated.

Keywords: Agility, lessons learned, DevOps, culture, scrum, XP, methodologies.

Resumen

El objetivo fue investigar las buenas prácticas que se deben seguir para implementar la cultura DevOps para acelerar el desarrollo de aplicaciones de software en periodos más cortos y unificar la historia para que puedan trabajar de la mano. La investigación fue bajo un enfoque cualitativo y una revisión de referencias para proporcionar información sobre los términos y metodologías de desarrollo utilizadas. Finalmente, para poner en práctica DevOps, es necesario entender que, aunque se tengan las mejores herramientas disponibles, si no se cuenta con la cultura, no se obtendrán mejores resultados ya que es una combinación de buenas prácticas para poder funcionar. En este sentido, para colaborar con este cambio de cultura, es fundamental el apoyo de líderes capacitados, que apoyen y apadrinen la nueva forma de trabajo, donde se puedan integrar todas las áreas de la empresa.

Palabras clave: Agilidad, lecciones aprendidas, DevOps, cultura, scrum, XP, metodologías.

Introduction

Software development has become almost a necessity in organizations because it allows the use of tools that make information available to employees in the performance of their work activities and make decisions according to actual data, improving response times and meeting the needs of their customers in a timelier manner. With time, methods have been presented that allow agile development, managing to accelerate the growth of software applications in shorter periods; this is because it is not enough to have an Information Technology (IT) department, but it is required to implement products quickly and with the ability to adapt to change [1]. With the emergence of functional outcomes in a short time, it should be facilitated, accelerated, and transferred to the production area repeatedly. So, it would be best to have a comprehensive risk control, with contingency plans, periodic reviews, and steps to follow if the risks materialize.

The emergence of agile development methodologies creates new challenges at the organizational level since product launches are produced in less time [1], generating more work for the operational area, accumulating a stored position, and awaiting attention to be transferred to the production area. This creates the need to look for strategies that help rearrange how organizations operate to avoid these barriers between departments.

For this reason, between the years of 2007 and 2008, the DevOps concept appeared [2]; it is a philosophy, an approach, and a change of culture based on good practices, which manage to remove the divisions that exist between the operations area and the IT development area so that they can work hand in hand, achieving greater productivity, promoting communication, the collaboration between people, feedback, and therefore, performs the automation and integration of the two areas [3]. This solution is implemented by large companies worldwide and has been proven to work [4].

Companies or institutions that wish to implement the DevOps culture must face significant challenges and measure if they are prepared for this change, in addition to continuously evaluating themselves to know if they are achieving the objectives this new way of working promises. It must be evident that this change only happens in the short term; it takes time, effort, and continuous improvement, in addition to the collaboration of all the people in the organization to optimize the results [5]. So, in this article, we investigated the methods, tips, and good practices that should be followed for its implementation, as the need to apply evaluations to work done by the teams to analyze whether or not communication, the collaboration between people, and other aspects with which you can check the achievement of the objectives of the organization when implementing DevOps.

With this organizational change, companies have an essential ally, which is the use of the tools already available in the market to implement DevOps that allows them to meet the requirements and needs of the business. However, for this research, the software designed by Microsoft Corporation [6], MS Team Foundation Server (TFS), will be taken as a case study to check the value it can provide to organizations and collaborators in this new need for change.

Additionally, it was analyzed whether, in a DevOps implementation, it is necessary to create new roles or positions in the organizations so that they can help coordinate, guide, or advise the new ways of working. We also analyzed the changes that must be implemented in the work cycles to unify the personnel of the different departments managing each development stage in another way. In the market, various tools collaborate with implementing DevOps in organizations of form agile developed to help companies accelerate the process of developing and delivering products and services, allowing them to reduce their time-to-market. Still, for this article, we only studied the TFS tool, its advantages, and whether it has facilities to cooperate with the change of culture.

Literature review

Software development has not been evolving in all organizations in the same way; some continue with traditional methodologies, where changes are difficult to implement, users do not have much interaction, have much documentation, are process-oriented, are very predictable, product delivery at the end of the project and sometimes cost estimates, time or development functionalities throughout the project, are no longer what was planned in the initial stages [7]. It is essential to mention that for a long time, these methodologies were successful; however, for other companies or institutions, these techniques no longer meet their needs, so they had to look for other, more agile methods.

Therefore, in 2001 a document known as the Agile Manifesto was created, where several principles are mentioned to develop software faster. Among the principles mentioned in the paper are: individuals, tools, and interactions over processes; working over extensive documentation; and response to change over following a plan [7]. With this, we can see that the vision of the way we were working had to change since we were emphasizing aspects that did not need so much effort, such as documentation, to have that time and plan more efficiently the constant changes that could occur according to the needs of the business in the different stages or phases of the project.

After this publication, different agile development methodologies appeared; Scrum and Extreme Programming (XP) are the most known. In each method, the ways of working are defined, each with its roles, work groups, and the same objective of developing high-quality software quickly that meets the goals of the user or customer [7].

The scrum methodology works with small groups of less than ten people (recommended); the interactions can last two to four weeks, according to the needs. It can be used in any project, regardless of size, and three roles are defined: Scrum Master (SM), Product Owner (PO), and Development Team (DT) [7].

The PO is accountable for making decisions about the product, providing value, and knowing the business; this role is intended to simplify communication, as it falls on a single person. The DT to develop the product; they are not distinguished with separate roles, instead what is sought is that the responsibility for the deliverable belongs to everyone; they are self-organized and collaborate; they respect each other's opinions and work for the same goal since everyone is essential. Finally, there is the role of the SM, who leads the project, collaborates with the communication between the Product Owner and the Development Team, and guides so that the Scrum rules comply with Menzinsky, López and Palacio [8].

In XP, people work in pairs, which seeks to reduce development costs and increase quality; interactions are from one to three weeks, and requirements are documented in detail as user stories and are given priority and estimated times. It is used in small projects; seven roles are defined: Coach, Programmer, Tracker, Customer, Consultant, Tester, and the Big Boss [9].

Among the most critical roles of this methodology, we have the Programmer, who is responsible for all the stages of product creation and can be said to be similar to the part of the Scrum Development Team. The customer is responsible for writing the user stories based on the requirements and verifying the advanced functionalities' testing. The Big Boss is in charge of project coordination, collaborates with the teams' training, and provides the resources needed by those teams [9].

So, both methodologies have succeeded in enabling development teams to deliver functional products that need to be put into production for customers to use more interactively. This is being implemented in many places where software developers can see the significant advantages that this achieves. Still, not all problems end there, since increasing the productivity of development also increases the traffic of requests to the operations area, who are responsible for putting these developments into production, which causes bottlenecks, so that a new case to be solved is in sight.

The people in charge of software development are in the development area (Dev/Development), and the operations area is the people putting the products into production and attending to the different user requests. They are also the ones who know the business best (Ops/Operations). The union of these two areas creates the word DevOps, which seeks the elimination of barriers between the regions, achieving greater visibility and

collaboration between them. With the teamwork of these two departments, more significant innovation is sought, managing regulatory restrictions more agilely. With this, it is about obtaining operational excellence with the high availability of the tools and the advantage of recovering faster [10].

With this new term, it is possible to give importance to the participation of people in organizations. With the above, there might be many automated tools and efficient processes, which are only helpful if people have collaborative attitudes and can execute efficiently. Achieving a DevOps culture is not an easy task; it requires constant communication between all stakeholders, leaders who monitor that an environment of synergy is being created and that there is an exchange of knowledge, in addition to removing barriers that may arise that interfere, and that there is a shared responsibility of the areas in achieving the objectives [11].

Now, to implement DevOps, the first thing is to understand what the business needs are, what is failing, or what can be improved; with this, you can have an accurate picture on which they can be based to raise the strategies to follow, and the goals they are looking for DevOps helps greatly in three areas: the customer experience is improved since it is attended or can take into account the feedback, which causes satisfaction; increases the capacity for innovation, achieving that increases the value, with this it is meant that by increasing productivity increases efficiency, changes the culture; and finally, the automation of application delivery is increased [11].

Among the techniques used in adopting DevOps is continuous improvement, in which lessons learned are considered, as well as the observations that are head of the processes and products that the organization has. Another of the techniques used is the planning of deliveries; this is due to the number of deliveries generated in short periods, so the aspects to be taken into account in each process must be studied without neglecting quality assurance. Also, there is continuous integration because each team member can work separately on a specific change. When it is finished, it should be able to join the project quickly and agile. Another aspect to consider is to give continuous follow-up and feedback, that is, to monitor what is in production in terms of performance and ensure that it meets the expectations for which it was created. Last but not least, constant testing ensures that the software and the integration are done transparently and with quality, preventing it from damaging what has already been created [11].

Thus, what is recommended to be able to implement DevOps is that all people should use the same tools to be able to achieve that all parties can have the same information that they have the same data since this helps or collaborates so that they can work in the same line, have visibility in the fulfillment of the objectives and make decisions with accurate and updated data. However, many times it is not possible to do it that way; that is, it is too late to use the same tools; for these cases, it is best to try to make a mix of all those that already exist with the use of components or the service of open-source interfaces [12].

In the market, several tools can be used to implement DevOps, which every day tries to integrate improvements to provide better control or collaboration with different areas of the organization. Some of them come from the technological giants that have incursions into solutions that help adopt the DevOps culture. Among the tools or solution providers are: Microsoft's Team Foundation Server tool, Atlassian's Jira Software solution, Amazon Web Services (AWS), IBM, Intel, and Google; open-source tools also emerge, which meet the objectives of being able to provide a collaborative solution [12].

Microsoft's Team Foundation Server (TFS) is a platform that can execute a vast number of functions, which allows interdisciplinary teams to integrate their work efficiently, and the size of the project does not impede since it is possible to work on any project regardless of its size [13].

One of the features of the TFS tool is that it can be used with many programming languages, e.g., Java, Python, Hyper Text Multi-Language release 5 (HTML 5), JavaScript, and C#, among others. This helps many organizations with some of these tools to be fine if they want to use them. Also, it can be used with Visual Studio, Eclipse, and xCode tools, plus it allows integration into any code editor (IDE). Therefore, it is not a tool with integration limitations since it has been improving so it can be used transparently [13].

Another use that the TFS tool has is for controlling code versions. In addition, it has unlimited capacity; it is private (permissions manage it to each repository), secure, and centralized [14]. With this feature, multiple people can work on the same version in different parts of the software, which helps each team member to upload their work, and others can get it efficiently by simply pressing the option to get the latest version. If two people have worked on the same component, the program prompts you to check if you want to merge the two jobs or decide which one to leave.

Teams using agile methodologies can use TFS since this set of tools can implement Kanban panels, in which the progress of each of the jobs can be seen, ensuring that all members of the updated group can see the flow and stay connected [15]. Another option offered is the Scrum dashboard, with which the team can conduct planning, daily, and retrospective meetings very efficiently. In addition, all users can see the code that is linked to each of the stories, in which each of the changes can be visualized, the errors that it presents, and the history of the evolution remains so that it can be tracked in case it is necessary [15].

Added to this, TFS collaborates with the integration and delivery of software products continuously; software can be deployed from anywhere and can be integrated at any time since it can work in the cloud. For this purpose, load testing can be performed, which allows testing of the infrastructure in place and the number of loads to be performed [15]. Integration with TFS can be done towards third parties, either applications or services, using open standards; some of the Application Program Interfaces (APIs) mentioned are REST and OAuth 2.0 [16].

Also, Microsoft offers the possibility to have an express version for free for users who use it individually or for small teams with less than five members. No server is needed for this version; you can take the stored history if you want to migrate to another paid version. So, having commented on the set of TFS tools and their features, this paper will analyze how to use them to implement the DevOps culture in organizations.

Methodology

This research has a qualitative approach because it aims to explore a theory and then turn to the empirical world to confirm if it is supported by the facts [16]. This is a case study review since it was based on compiling and discarding what is not necessary and reviewing the references of other works done to provide information on the terms used [17]. These development methodologies are currently employed; in any case, it is essential that the method used is clear and rigorous to guarantee the reliability and validity of results, in addition to the search for the tools that are exposed in this research [18-20].

In terms of scope, this article covered a descriptive approach to the characteristics of Scrum, XP methodologies, DevOps culture, and the functions of the TFS tool. Finally, we proceeded to analyze how DevOps can be implemented in organizations with the help of the tool.

Results and discussion

The cycles performed in agile methodologies seek the delivery of potentially usable products, which means that they can be placed in production environments so that the end user can use them; of course, for this to happen, it must first be taken into account that the software has successfully passed all phases, including being tested promptly so that it would go to the operations area for its implementation field [5, 21].

Thus, to achieve that, usable products are generated quickly in agile methodologies; for example, Scrum works with several ceremonies, which streamline processes. In addition, it gives the possibility to ask for new requirements or make changes to the needs that are allowed without causing significant problems [10].

Among the ceremonies implemented in Scrum is the product stack, which is the customer requirements; the customer prioritizes them and can change that when the business merits it. This list is not definitive and can constantly change [8]. With this list, meetings are held to plan each Sprint to be worked on, in which the

tasks that can be performed according to the team's capacity and the time defined, which can be from 2 to 4 weeks, are listed. At the end of the Sprints, a retrospective must be held, in which aspects that can be improved are sought. Daily the Scrum team conducts follow-up meetings of the activities, in which they see what they did yesterday, what they will do today, and finally if they had any difficulties or problems [10].

The requirements are presented in a tool that must be updated according to work performed, and if priorities change, everyone must be aware of such changes; this list is called a product backlog.

In organizations where many disciplinary groups can make quick deliveries of products to the operations area, this causes bottlenecks since they will need more time to meet all requests. This results in customers not being able to use the deliverables once they have been developed, which is of no benefit to the customers, much less to the business in general, but instead causes losses and pressure on the different areas.

The main idea of DevOps is to reduce the time in the processes or software development cycles so that end users can get the product of value as soon as possible so that it can be used for the benefit of the business and they can get their advantages in the earliest way [22]. When the customer receives the software in production, ready to be used, he can obtain benefits since he has tools that help him to perform his work in a better way; even in some cases, he can measure his performance with the data that is being obtained, he can see which one or which of his processes can be improved, and try to implement a strategy to achieve it. There are many reasons why a business needs to have all its assets working, as in the case of software, which is of no benefit if it is only in the queue to be implemented.

DevOps cycles change from other bikes to the extent that their duration changes; this means that all processes done manually can be automated, lowering their duration time significantly [23]. Therefore, the stages that must follow their products, such as planning the product to be made, generating the code, testing the code, releasing it, implementing it, operating it, and monitoring it, and what changes with DevOps, is to do this in an agile way, and constantly. It is done repeatedly, and when the cycle is finished, it starts again, trying to improve the duration of the processes in practice.

This is where DevOps cycles come in, which seek to ensure that the entire process from system development to production is carried out quickly. This, with the collaboration of both the people who are in development and the people who are in operations, where the communication channel can be used to achieve this, with regular meetings, where people are aware of all possible changes that can be implemented and what aspects must be taken into account to avoid problems.

One of the processes that can be improved with the implementation of DevOps is the use of Continuous Integration (CI), which is the agility to add more code to the solution or modify it when a corresponding change request has been made. When a code is generated, it can be deposited in what is known as version control, where you can have several versions, in which each contemplates the changes or improvements separated securely and where they can be backed up. The idea is that each person in the project can generate their code in isolation. Once they finish their development, they can integrate it into what is known as the complete main branch, where the already integrated and tested code of the whole team is [24].

Before, this was extremely difficult to do since everyone worked on their code locally. Trying to unite it after several weeks, with all the advances of each one, was a real headache since myriad problems arose. For this reason, using version control can avoid these problems and help reduce development times since it is automated.

Before, this was extremely difficult to do since everyone worked on their code locally. Trying to unite it after several weeks, with all the advances of each one, was a real headache since endless problems arose. For this reason, using version control can avoid these problems and help reduce development times since it is automated.

Another practice that can be used for DevOps is Continuous Delivery (CD), which is the entire process used from creating or constructing the software until it is taken to production. Now, to have control of this process, it is necessary to have approval in the passage of each of the stages of the process so that they can be

easily auditable and comply with the methods already defined in the organizations as regulatory. If this process is done manually, it will cause errors and delays in the delivery of the products. As for the validation performed in this process, the tests with the highest probability of failure are implemented first; of course, automated tests can be contemplated for this to be agile. After the most probable tests have been performed and completed, longer and more complex tests can be exercised [25].

Several tests must be performed depending on the business needs and not all of them can be performed automatically; there are some that, due to their process, need a pair of eyes to check their proper functioning. The CD process contemplates building, testing, and deploying what is being worked on in a production environment. Then, to move on to the other stages, these must be completed correctly to move on to the next one, to take the work to production with the least number of errors.

Here also enters the agile planning which is the use of frameworks to produce the codes or deliverables faster, with the support of agile methodologies, which accelerate product delivery, where the team works towards the project objectives. All have the same responsibilities to achieve the planned goals [26].

Monitoring or registering running applications is constantly reviewing the performance of the developments that reached production to verify that they are running according to expectations. This is because the DevOps experience does not end when the requirement is put into production but continues with the verification of the products at all times since, depending on the use given by the users, they can check if the application is responding as desired. In the event of errors or any problems, these are detected immediately and are treated so that the user experience is not harmed. Then the root of the problem is sought to find the solution and prevent it from happening again [27].

Another aspect of good practices that should be implemented with DevOps is the application of tests, which can be manual, where you can use a human resource or several, or automated tests that are performed with the help of a robot that performs the number of times necessary to complete them successfully, these tests can be a black box or white box [10].

Communication and feedback are fundamental in the DevOps culture since they must exist between people in development and those in operations. This is to avoid future conflicts or different configurations between the versions used by the test or development environments and the version to be implemented in the production [28]. What is proposed are periodic meetings so that the operations people can give more feedback on what they know about the business, talk about what is being worked on, and work hand in hand with both areas. Both areas should know about DevOps to understand that barriers between both departments should not exist. You can even evaluate processes between products and other issues that help ensure that what you develop meets the business's and customers' expectations.

DevOps is not a role or a position; it is a culture and good practices, so it cannot be said that you need to create functions to implement it. However, to be part of this new way of working, the person must have some interpersonal skills or techniques, in addition to possessing knowledge of the organization's business processes, operational practices and software issues, and infrastructure with which the business has [27].

In some articles, it is mentioned that the person who should be in charge of leading DevOps strategies is the Chief Information Officer (CIO); if this is to be carried out satisfactorily since his sponsorship is needed, the changes are lasting and are taken by all as a practice, it would also collaborate between the different teams. With this, what is sought is that all stakeholders are involved, which would help facilitate agile management in the demand of requests for the development team [28].

Currently, several tools in the market collaborate to implement DevOps, but as mentioned above, this research will focus on using the Team Foundation Server (TFS) tool. The application that Microsoft will use to implement TFS is Visual Studio. It can be created in several platforms such as Visual Basic, Visual F#, Visual C#, Visual C++, ASP.Net, and JavaScript; it can also be used online.

To configure TFS in Visual Studio, you must have a server that has TFS configured; this is in case you will use more than five users. In the case of fewer than six people, Microsoft gives the option to use a shared server,

and when you want to move to a private server, you can do it by migrating the information that has already been worked on. Well, entering the server information is done straightforwardly.

So, to download the projects, you must wait since it will last according to the number of projects you want to connect to and the amount of information stored for each project. With TFS, it is possible to manage agile projects since it is possible to create requirements and, from each of them, complete tasks assigned to each work team member. Thus, the status and changes that are being executed are tracked with the help of the tool's dashboard. For this purpose, TFS has a Dashboard with a "Backlog" to work on each Scrum team iteration. The tasks are ordered according to priorities, and assigning that task to an owner is also available. Each person can allocate the time or effort they deem appropriate to finish the job [29].

With TFS, one can have version control since the project is stored on a server, and everyone can download the code and documents generated over time. There are two ways of working: one is online or connected to the server where the application is configured; the other way is when there are network problems, and the connection to the server cannot be made, or when other cases arise that prevent such contact, it is possible to work locally [29]. For the first option of working with connection, it is possible to see if someone else is working on the files to be worked on, or if not, if they are free.

TFS provides the necessary information to see if two or more people are working on the duplicate files, and if so, you can merge the different jobs or decide which version to keep for uploading. Also, the application offers the option of consulting the change history, which shows the change made to the file, the user who made the change, and the date the change was made.

When it wants to release new products inside the project, the ideal is to make a Branch of the project so that you can have the previous version in the same form that you have, for example, in production, and work on another version with the changes or improvements that request personal information from the user according to the needs of the business. With this branch, the history of changes made to each file is preserved [30].

In this Branch creation option, it is possible to work in different ways. For example, you can have the main branch, which is the complete work or what has been done so far, and the need arises to create two new changes, which would do it for two different resources, in which case each one would have its branch. Then, when each one finishes its work, these can be joined to a new Branch named "Development" to see if there are conflicts and so on in the union; the cases that arise are worked on, and once they are ready, the tests are applied. Once completed, they would form a new version of the project [30]. These are some of the facilities provided by using the application.

When a file is being edited in the solution explorer, it is shown if it is being modified with the check symbol ✓; in case a new file has been added, it is displayed with the plus sign +; also, in case it has not been touched it is shown with the padlock icon. Also, when another person works on a file, it informs with the user icon. It can be configured to display deleted files; these would be shown with a red X. With all these symbols or icons, you can see how the TFS tries to inform about the activities being done in each of the files the project has.

In the Team Explorer tab, as shown in Figure 1, you can see what is pending. This means that the files that have been modified or added will be displayed. This should be done with the Check-In option when you want to upload a file. Still, to control that only what is ready is uploaded, you can see that TFS separates the things to be uploaded in the Included Changes section, and the files left out will be in the Excluded Changes option. In addition, it also shows the number of files that are in each of the areas [31]. To upload the changes, it is recommended that in the Comment section, you write down what was done in the file or files so that in the history, you can summarize what was done in that change.

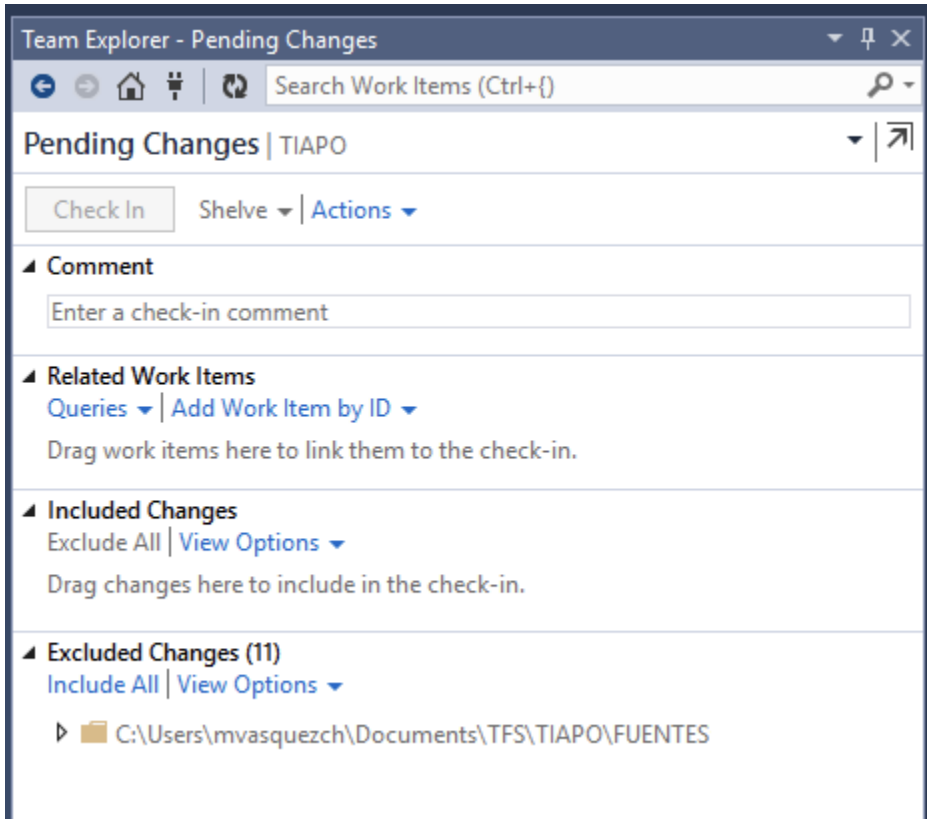

Figure 1
Pending changes Team Foundation Server

So, TFS also gives the option to see what was done in each of the files with the compare function since it can see the differences in two files, even if they are the differences that are in the local file, with the one on the server [32].

For the execution of the tests, Microsoft has created the IntelliTest component, which facilitates the application of white box tests, which is configured in Visual Studio, and tests can be applied to the method that you want to use; it also facilitates the solution because it shows a detail of the error. To execute the tests, it is positioned in the method, given right click on the way that you want to test in the option, and finally, given a chance to execute. It is executed several times with different inputs when this option is pressed, showing which values were entered and the results. Also, you can test an entire class the same way the method is positioned over the type and repeat the same action. This functionality also allows storing the executed tests in a separate project [32].

Conclusions

The emergence of agile development methodologies for IT projects has achieved its goal of creating usable software products in short periods, with the flexibility that the requirements can be easily changed and can change their priorities at any time, ending the roles in the development area, since it manages to unify the responsibility for the entire team.

This acceleration in the development area has caused stagnation in the operations area, which has received more passes to production in short periods. DevOps is a strategy that seeks to reduce user product delivery times, collaborating with the development and operations areas without neglecting the business and all parties involved. Thus, it aims to create a culture in the organization where cooperation, communication, and

feedback are the allies so that they can have a better time developing products and achieving competitive advantage and satisfaction of customers or end users.

In this sense, to collaborate with this change of culture, it is essential to have the support of trained leaders, who support and give sponsorship to the new way of working, where all areas of the company can be integrated.

The use of tools is a perfect strategy to reduce the time between the stages of product creation, as is the case of the TFS application, which can be implemented in various programming languages, which is a great advantage since it can be integrated into the software of many companies that already have their software created.

TFS is a potent tool in which you can work with version control in isolation, and when you have the finished requirement, you can easily unify the project. It can work by versions where the histories are saved, and the tests can be done before joining it to the main branch. In addition, it is possible to reduce the time for the continuous delivery of products.

The implementation of DevOps is a change in the way of working since it encourages teamwork, which avoids the barriers between departments so that they can work hand in hand to achieve benefits in having functional software that can take advantage of the end users.

Finally, to implement DevOps, it is necessary to understand that, even if you have the best tools at your disposal, if you do not have the culture, you will not achieve better results since it is a combination of all the best practices to be able to work. In addition, it is a change that always continues because you must always be aware of continuous improvements consistently to achieve improvements in processes and products. The evolution of DevOps in business is a new frontier in creativity and teamwork, playing together to build a more competitive organization.

For future lines of research, it is proposed to review the existence of success stories with the implementation of DevOps using the TFS tool in Costa Rica or any Latin American country to apply interviews about the experience they have had with its performance with this tool, also how it has changed the culture of the organization, to know which are the people responsible for ensuring compliance with the change of the organization and how success is measured in the organization with the use of these good practices.

Additionally, it is recommended to study the new trends in the market related to the DevOps paradigm and the focus on reducing software development cycle times to obtain products that can be used in short periods to satisfy the needs of the business.

References

[1] O. Salo and P. Abrahamsson, "Agile methods in European embedded software development organizations: a survey on the actual use and usefulness of Extreme Programming and Scrum," IET software, vol. 2, no. 1, pp. 58-64, 2008, doi: https://doi.org/10.54489/ijtim.v2i2.77.

[2] M. S. Khan, A. W. Khan, F. Khan, M. A. Khan, and T. K. Whangbo, "Critical challenges to adopt DevOps culture in software organizations: A systematic review," IEEE Access, vol. 10, pp. 14339-14349, 2022, doi: https://doi.org/10.1109/ACCESS.2022.3145970.

[3] M. Gall and F. Pigni, "Taking DevOps mainstream: a critical review and conceptual framework," European Journal of Information Systems, vol. 31, no. 5, pp. 548-567, 2022, doi: https://doi.org/10.1080/0960085X.2021.1997100.

[4] P. Dora. "State of DevOps Report." Puppet. http://www.berrykersten.nl/wp-content/uploads/2017-state-of-devops-report.pdf (accessed 2022).

[5] M. Senapathi, J. Buchan, and H. Osman, "DevOps capabilities, practices, and challenges: Insights from a case study," in Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018, 2018, pp. 57-67, doi: https://doi.org/10.1145/3210459.3210465.

[6] Microsoft. "DevOps overview for VSTS and TFS." Microsoft. https://docs.microsoft.com/en-us/azure/devops/user-guide/devops-alm-overview?view=azure-devops (accessed 2022).

[7] T. Dingsøyr, S. Nerur, V. Balijepally, and N. B. Moe, "A decade of agile methodologies: Towards explaining agile software development," vol. 85, no. 6, pp. 1213-1221, 2012, doi: https://doi.org/10.1016/j.jss.2012.02.033.

[8] A. Menzinsky, G. López, and J. Palacio. "Scrum Manager v2.6." Scrum Manager. https://www.academia.edu/30510102/Scrum_Manager_v_2_6 (accessed 2022).

[9] F. Anwer, M. Aftab, and S. Syed, "Comparative analysis of two popular agile process models: Extreme Programming and Scrum," International Journal of Computer Science and Telecommunications, vol. 8, no. 2, 2017. [Online]. Available: https://www.ijcst.org/Volume8/Issue2/p1_8_2.pdf.

[10] M. D. Kahya and Ç. Seneler, "Geographical distance challenges in distributed agile software development: Case study of a global company," in 2018 3rd International Conference on Computer Science and Engineering (UBMK), 2018: IEEE, pp. 78-83, doi: https://doi.org/10.1109/UBMK.2018.8566591.

[11] R. Jabbari, N. bin Ali, K. Petersen, and B. Tanveer, "What is DevOps? A systematic mapping study on definitions and practices," in Proceedings of the scientific workshop proceedings of XP2016, 2016, pp. 1-11, doi: https://doi.org/10.1145/2962695.2962707.

[12] H. Admin. "Tooling up for DevOps." HDWebsoft. https://www.hdwebsoft.com/blog/tooling-up-for-devops.html (accessed 2022).

[13] S. Neely and S. Stolt, "Continuous delivery? easy! just change everything (well, maybe it is not that easy)," 2013 Agile Conference, pp. 121-128, 2013, doi: https://doi.org/10.1109/AGILE.2013.17.

[14] F. Medina-Domínguez, M.-I. Sanchez-Segura, A. Amescua, and J. García, "Extending Microsoft team foundation server architecture to support collaborative product patterns," Software Process Dynamics and Agility: International Conference on Software Process, ICSP 2007, Minneapolis, MN, USA, May 19-20, 2007. Proceedings, pp. 1-11, 2007, doi: https://doi.org/10.1007/978-3-540-72426-1_1.

[15] G. Azizyan, M. K. Magarian, and M. Kajko-Matsson, "Survey of agile tool usage and needs," 2011 Agile Conference, pp. 29-38, 2011, doi: https://doi.org/10.1109/AGILE.2011.30.

[16] J. Ofoeda, R. Boateng, and J. Effah, "Application programming interface (API) research: A review of the past to inform the future," International Journal of Enterprise Information Systems (IJEIS), vol. 15, no. 3, pp. 76-95, 2019, doi: https://doi.org/10.4018/IJEIS.2019070105.

[17] J. C. Greene, J. L. Compton, E. Whitmore, and H. Sappington, "Strategies for qualitative data analysis," Evaluation Practice, vol. 8, no. 4, pp. 5-11, 1987, doi: https://doi.org/10.1177/109821408700800401.

[18] A. Queirós, D. Faria, and F. Almeida, "Strengths and limitations of qualitative and quantitative research methods," European Journal of Education Studies, 2017, doi: http://dx.doi.org/10.46827/ejes.v0i0.1017.

[19] A. Shah, G. Piro, L. Grieco, and G. Boggia, "A qualitative cross-comparison of emerging technologies for software-defined systems," in 2019 Sixth International Conference on Software Defined Systems (SDS), 2019: IEEE, pp. 138-145, doi: https://doi.org/10.1109/SDS.2019.8768566.

[20] H. Taherdoost, "What are different research approaches? Comprehensive review of qualitative, quantitative, and mixed method research, their applications, types, and limitations," Journal of Management Science & Engineering Research, vol. 5, no. 1, pp. 53-63, 2022, doi: https://doi.org/10.30564/jmser.v5i1.4538.

[21] L. E. Lwakatare et al., "DevOps in practice: A multiple case study of five companies," Information and Software Technology, vol. 114, pp. 217-230, 2019, doi: https://doi.org/10.1016/j.infsof.2019.06.010.

[22] W. de Kort, "What Is DevOps?," DevOps on the Microsoft Stack, pp. 3-8, 2016, doi: https://doi.org/10.1007/978-1-4842-1446-6_1.

[23] L. Zhu, L. Bass, and G. Champlin-Scharff, "DevOps and its practices," IEEE Software, vol. 33, no. 3, pp. 32-34, 2016, doi: https://doi.org/10.1109/MS.2016.81.

[24] M. Shahin, M. A. Babar, and L. Zhu, "Continuous integration, delivery and deployment: a systematic review on approaches, tools, challenges and practices," IEEE Access, vol. 5, pp. 3909-3943, 2017, doi: https://doi.org/10.1109/ACCESS.2017.2685629.

[25] L. Chen, "Continuous delivery: overcoming adoption challenges," Journal of Systems and Software, vol. 128, pp. 72-86, 2017, doi: https://doi.org/10.1016/j.jss.2017.02.013.

[26] P. Gould, "What is agility?," Manufacturing Engineer, vol. 76, no. 1, pp. 28-31, 1997, doi: https://doi.org/10.1049/me:19970113

[27] T. Schlossnagle, "Monitoring in a DevOps world," Communications of the ACM, vol. 61, no. 3, pp. 58-61, 2018, doi: https://doi.org/10.1145/3168505.

[28] C. Jones, "A proposal for integrating DevOps into software engineering curricula," in Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment: First International Workshop, DEVOPS 2018, Chateau de Villebrumier, France, March 5-6, 2018, Revised Selected Papers 1, 2019: Springer, pp. 33-47, doi: https://doi.org/10.1007/978-3-030-06019-0_3.

[29] W. De Kort, DevOps on the Microsoft Stack. Springer, 2016.

[30] J. Palermo, ".NET DevOps for Azure," Apress Berkeley, CA, vol. XVII, p. 272, 2019, doi: https://doi.org/10.1007/978-1-4842-5343-4.

[31] J. Levinson and D. Nelson, "Team Foundation Version Control," Pro Visual Studio 2005 Team System, pp. 59-102, 2006, doi: https://doi.org/10.1007/978-1-4302-0171-7_3.

[32] D. Ivanov et al., "UnitTestBot: Automated unit test generation for C code in integrated development environments," 2023 IEEE/ACM 45th International Conference on Software Engineering: Companion

Proceedings (ICSE-Companion), pp. 380-384, 2023, doi: https://doi.org/10.1109/ICSE-Companion58688.2023.00107.

Gabriel Silva-Atencio, Mauricio Umaña-Ramírez

Evolution of DevOps: Lessons learned for success as part of digital strategy

Evolución de DevOps: Lecciones aprendidas para el éxito como parte de la estrategia digital

Tecnología en marcha
vol. 37, núm. 2, p. 23 - 35, 2024
Instituto Tecnológico de Costa Rica, Costa Rica
revistatm@itcr.ac.cr

/ ISSN-E: 2215-3241