

A STUDY ON DEVELOPMENT OF SOFTWARE APPLICATIONS USING EXTREME PROGRAMMING AND DEVOPS

Mrs. Nagalambika*
Dr. L. Manjunath Rao**

ABSTRACT

To be competitive in today's market, every software company has modified its business strategy to compete at the highest level. As a result, new software is developed fast to take advantage of new opportunities. It quickly adapts to new technology, tools, and processes. The purpose of this article is to study, analyze, and implement the Extreme Programming (XP) methodology with a DevOps culture. In today's software business, Extreme programming is a well-known Agile methodology that adheres to a set of principles and values. The study involves a case study methodology and data collection from industry experts through one-on-one interviews, and all experts are implementing a DevOps culture on multiple projects. As a result, the combination of XP methodology and DevOps culture aims to increase the quality of software code and, through automation, it maximize efficiency while delivering software. It puts a greater emphasis on operational and business preparedness. Along with automation and early issue detection, it helps to improve quality. This approach allows automating on-demand software deployment to any environment.

KEYWORDS: *Agile Methodology, Extreme Programming, DevOps, Continuous Integration, Software Applications.*

Introduction

Agile development is a broad phrase that refers to a variety of iterative and incremental software development methods. This helps the project to rapidly adapt to changes and focuses on delivering the smallest working piece of functionality as soon as feasible. Throughout the project's life cycle, we'll keep improving it and adding new features. Extreme Programming, Scrum, Kanban, Crystal, and Rapid Application Development are some of the important agile techniques.

Extreme Programming (XP) is a software development approach. Strong customer integration, thorough testing, code-centric development, documentation, refactoring, and paired programming are all emphasized. XP's goal is to help software engineers accept change and cope successfully with the problem of unclear and evolving requirements. Due to its distinctive lightweight techniques, XP is one of several agile processes that distinguishes itself from more traditional ways.

* Research Scholar, Department of MCA, Dr. Ambedkar Institute of Technology, Bangalore, Karnataka, India.

** Professor and Head, Department of MCA, Dr. Ambedkar Institute of Technology, Bangalore, Karnataka, India.



Figure 1: Extreme Programming

Continuous integration is an agile engineering approach that stems from Extreme Programming. Although the words have been around for a while, DevOps has embraced them since continuous integration requires automation. Continuous integration is frequently the first step on the DevOps maturity road.

WHY DevOps?

The Agile approach concentrates solely on developing functional software, and it ignores how the software will be affected by the live production environment, as well as how the programme will be affected by the live environment. This technique, which excludes operations from the software development process, causes issues and conflict between the development and operations teams.

This is because it's difficult to tell if an issue is caused by the code or the deployment when anything goes wrong. The result is a lot of pointing of fingers between the two departments. People in the software development industry eventually became irritated with the divide between development and operations. As a result, a new method evolved, removing the barrier between the Dev and Ops teams, and DevOps was born [7].

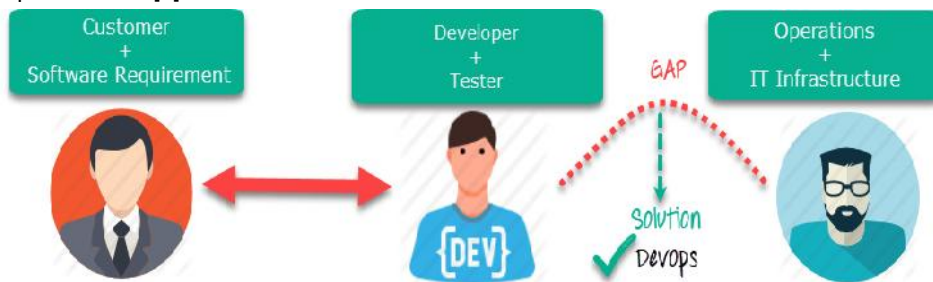


Figure 2: DevOps Solution

What is DevOps? and Its Practices

DevOps is a software development and delivery method that stresses communication and cooperation between product management, software development, and operations experts. It's a mash-up of the words "development" and "operations". It accomplishes this through automating and monitoring the software integration, testing, deployment, and infrastructure modifications processes, as well as by fostering a culture and environment in which software development, testing, and release may occur more quickly, often, and reliably [10].

DevOps is a collection of techniques aimed at shortening the time required to update a system and deploy it in production. It puts a greater emphasis on operational and business preparedness. A DevOps practise is any approach that helps you achieve these aims [5].

The following are some of the DevOps practices:

- Continuous integration is a DevOps recommended practice that involves developers merging code changes into a codebase that undergoes automated builds and testing. You avoid the integration issues that might arise if you wait until release day to integrate modifications into the release branch.

- Continuous delivery is similar to continuous integration in that it automatically delivers all code modifications to a testing and production environment, completing the build phase.
- Continuous deployment is a step above continuous delivery. This approach is used to deliver every update that covers all phases of the production process to your clients. Human interaction is not required, and just a failing test will prohibit a new modification from going into production. Below, figure 3 shows how the practices relate to each other.

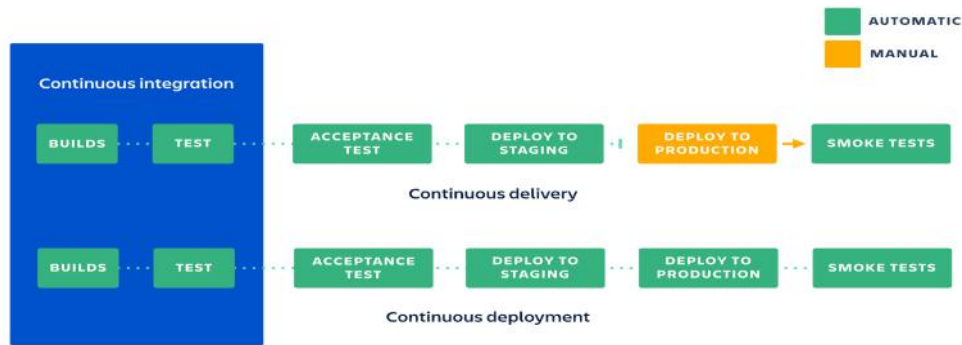


Figure 3: Continuous Integration, Delivery and Deployment

Related Work

According to a recent Suse study, 86 percent of IT leaders (1400) view DevOps as a part of their future IT strategy, with 77 percent planning to include DevOps approach into application development and delivery[7].

The DevOps idea was created to fill the gap between the creation of software and its deployment into production in major software organisations. DevOps' major goal is to use continuous software development methods [8].

The software development industry has changed as a result of new development standards and service delivery methods. DevOps has brought Agile methodologies to a conclusion, resulting in higher software quality as well as increasing delivery speed[1].

By utilizing rapid (agile) development cycles, quick (cloud and virtualized) provisioning of core infrastructure or platforms, improved communication between stakeholders and clients, application developers & operations engineers, and tools and methods, DevOps enables faster release cycles[2].

Research Method

We used a survey questionnaire to gather data from industry experts through a series of questionnaires administered over a three-month period, and we received a large response from all designated roles involved in the development and execution of software applications, such as QA Manager, System Analyst, Advisor IT Architecture, Delivery Manager, Individual Contributor, and many others.

All of the industry professionals are from different companies, and they have shared their opinions and responses on a range of initiatives, based on their knowledge and expertise.

Result

The project that uses the legacy method takes three months for each release, and it's very difficult to track down errors, whereas the project that uses XP and DevOps practices together takes 10 to 21 days for each release, and it focuses on taking ready-to-release software and deliver it in a secure and reliable manner.

The major objective is automation, which works on the premise of maximizing efficiency while delivering software. It puts a greater emphasis on operational and business preparedness. Along with automation and early issue detection, it helps to improve quality. To maintain quality standards, developers must adhere to coding and architectural best practices.

Conclusion

XP approach emphasize on cross-functional cooperation, client evaluations, and feedback and regular releases. DevOps, on the other hand, is a paradigm that brings together developers and operations teams to ensure continuous testing and delivery.

The goal of combining the Extreme Programming methodology with DevOps practices is to increase the quality of software code, delivery speed and to shorten the time between a system change and the transfer of that change to the production environment. This approach allows automating on-demand software deployment, to any environment.

References

1. Luciano de., & Aguiar Monteiro.(2021). A Proposal to Systematize Introducing DevOps into the Software Development Process. *IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 269-291. DOI 10.1109/ICSE-Companion52605.2021.00124.
2. DevOps whitepaper, DevOps – Are we there yet? , <http://possible.mindtree.com/rs/574-LHH-431/images/devops-are-we-there-yet.pdf>, (2021,Oct 12).
3. F.M.A. Erich., C. Amrit,& M. Daneva.(2017). A Qualitative Study of DevOps Usage in Practice. *Journal Of Software: Evolution And Process*, DOI: 10.1002/smr.
4. Nikhil Govil., Mayank Saurakhia., Pradyumna Agnihotri., Sachin Shukla., & Shivam Agarwal. (2020). Analyzing the Behaviour of Applying Agile Methodologies & DevOps Culture in e-Commerce Web Application. *4th International Conference on Trends in Electronics and Informatics (ICOEI)*(48184), 899-902.
5. Armin Balalaie., Abbas Heydarnoori., & Pooyan Jamshidi.(2016). Microservices Architecture Enables DeOps: Migration to a Cloud-Native Architecture. *IEEE Software*, 33(3), 42-52.
6. Barry Snyder., & Bill Curtis.(2018). Using Analytics to Guide Improvement during an Agile-DevOps Transformation, *IEEE Software*, 35(1), 78-83.
7. <https://jumpcloud.com/blog/what-is-devops-why-important>, (2021,Oct 12).
8. Senapathi.M., Buchan.J., & H. Osman.(2018). DevOps capabilities practices and challenges: Insights from a case study, *Proc. of the 22nd International Conference on Evaluation and Assessment in Software Engineering*, 57-67.
9. <https://www.atlassian.com/continuous-delivery/principles/continuous-integration-vs-delivery-vs-deployment>, (2021,Oct 12).
10. Brandon T. Klein., Gerald J. Giese., Jayson C. Lane., John G. Miner., & John J. Jones. (2020). An Approach to DevOps and Microservices, *SANDIA Report*. <https://www.osti.gov/servlets/purl/1635752>.

