IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

How DevOps Enhances the Software Dévelopment Quality

Ravi Teja Yarlagadda
DevOps SME & Department of Information Technologyy,
USA

Abstract

Quality is a vital success factor that impacts the success of an organization as well as its products. The need for quality is a significant factor that impacts software development processes across multiple stages. DevOps is communications of the development and operations of the word, which is a model that can be used to enhance the quality of software. DevOps provide organizations with various benefits other than the potential to enhance software quality as it was built to enhance cooperation between the development and operations teams. This paper aims to define DevOps while investigating how the use of this model impacts software quality. Further, the paper also identifies various ways an organization can implement to enhance software quality continuously. A way of finding information in this study is conducting literature that will provide more information on current activities involving DevOps in the software development industry. This study found that through the use of the CAMS framework, systems and software quality is improved through practicing DevOps. In summary, when an organization aims to generate high quality software, it is best to implement and use DevOps.

Keywords : *Software Quality, DevOps, SDLC*

How DevOps Can Enhance Software Development Quality

I. Introduction

DevOps is a combination of the development and operations of the word, which utilizes both tools and practices designed to enhance a company's ability to deliver services and applications quicker and more effectively than other conventional software development processes. This method's speed usually allows an organization to serve its customers better while competing more efficiently within the market. Therefore, DevOps is about doing away with traditional complex and rigid teams, operations, and development. With this model, operations and development teams usually work collaboratively across the entire software development life cycle from development to testing and operations to deployment.

DevOps is a title given to a theoretical framework that, when properly implemented, provides a seamless and cohesive functioning of the operations and development teams within the company. For this model to work, it needs and consequently creates a culture of information and knowledge sharing that leads and enhances the collaboration between various teams in the company. Practicing DevOps' principles is usually beneficial for the software operations and development's performance. Its practice usually also has a very positive impact on web service development and quality assurance performance (Zhu et al., 2016). DevOps is included in the disciplined agile delivery, which is an established software development process. This model helps in establishing as well as achieving an effortless continuous delivery process. Therefore, DevOps can help in enhancing the software development quality.

Various practices help in the formulation of a strong DevOps culture. These include transparency, agile decision-making, integration process, consideration of perspectives, as well as flexibility. For consideration of perspectives, there is a need to consider a cultural change from the participants' perceptive and the driver. With flexibility without exceptional situations, a company must be adequately flexible to even temporally do away with DevOps values. Further, for this model to work, there is a need for an implemented proves to integrate all the changes occurring. The teams must also have the needed agility to decide on the tools to utilize based on the available expertise and skills. Further, both development and operations teams should have total transparency meaning there are no hidden agendas.

A DevOps culture is best illustrated as a method of responsibility sharing, open communications, mutual trust, and respect. If these teams are not managed optimally, it can negatively impact the service quality, software quality, and productivity. These teams need to work collaboratively within an environment where testing mobile software requires testing from several unique devices and varying operating systems, which can prove a challenge requiring the implementation of a new approach. Utilizing the DevOps approach ensures that both operations and development teams are one another's supplier and customer. The development team usually needs the operations in managing systems for large-scale information systems, and at the same time, the operations team depends on help from development teams to develop applications and tools for the operating systems. The development team also helps in enhancing securing, stability, as well as performance. Basically, this dependency between the teams usually creates the basis on which the DevOps' bridge can be founded. DevOps practices can be classified into various concepts like services, quality assurance, standards, and structures.

In information systems that link the operations, development, and customer support teams with the client, DevOps can significantly contribute to the quality assurance area. Quality assurance is challenging to predict, and the DevOps model improves this process by bringing the various stakeholders closer through tooling and cooperation. DevOps also allows for increased opportunities for collecting more data when compared to times in the past. Team members performing both operations and development tasks can also be responsible for providing quality assurance. Through implementing behavior-based operations, the varying roles within quality assurance can dictate how the system must deliver the desired outcomes and goals. DevOps also has a contribution to services. These principles are also advantageous within the service management frameworks as services heavily rely on cooperation between development and operations personnel. Integrating development and operations tasks and activities usually force a company to reassess its business model. Companies occasionally need to reassess their business models to utilize service-based models such as software as a service, and since DevOps has various competitive advantages, it provides immense value as a vital part of the new models (Lwakatare et al., 2015). In the past, organizations owned resources, whereas today, there are offered to them as services such as infrastructure and the platform. Therefore, supporting DevOps principles within the service management framework can prove beneficial since services majorly rely on seamless cooperation between development and operations teams.

DevOps also contributed to the information system development. This model has brought many significant changes in information system development. DevOps helps minimize the present gap between the end-user, operations, and developers, which allows for problem detection earlier than with other conventional models. Before, every scrum print software would function as expected full-filling specifications; however, the actual end-user would not validate the specifications. With DevOps implementation, we can utilize continuous development, which allows for software to be released to end-user more frequently. DevOps also allows for operations and developers teams to work collaboratively and more effectively and efficiently.

The implementation of DevOps has several benefits, including enhanced collaboration, reliability, raid delivery, speed, and security. DevOps practices usually allow an organization to move at the speed they require to adapt to changing markets more quickly, innovate faster, and become more efficient and effective at reaching business goals and objectives. DevOps races such as continuous delivery and continuous integration can ensure the quality of changes in the infrastructure and software updates, allowing an organization to deliver at a more rapid speed while ensuring end users have an optimum experience. Further, under this model, operations and developer teams usually work closely while combining their workflows and sharing responsibilities. Additionally, an organization can implement this model without neglecting its security through adopting integrated and automated security testing tools. This paper aims to investigate and present the relationship between DevOps practices and principles and enhancement of software quality in terms of technologies and instruments that allow for continuous delivery and integration, high-speed deployment, and test automation.

II. Literature Review

This section presents various studies on DevOps and elaborates on how these studies are related to the study. DevOps shows how the software production and information technology organizations work collaboratively to minimize the market time. The use of DevOps has become a high profile with markets such that the performance of the organizations that implement the model has been studied and measured.

DevOps can help an organization to reduce the time for marketing their software products. DevOps allows for the fast delivery of upgrades to features and even modifications. Software development teams usually interact with information technology operations to submit applications for testing. With the new software, there is a need for a test environment to be developed for the operations team. When the software is an enhanced and reviewed version of the current one, the operations team will require implementing and modifying the side applications and interfaces (Gill et al., 2010). As such, when activity team members test software, a developer can only deploy new code with the application. According to Nybom et al., 2006, DevOps usually eliminates the above time-consuming activities during software creation. For example, with automated software testing, feedback is provided rapidly, and automated integration allows for change to be made to code mush easily. As such, organizations can release small and even incremental software updates while providing customers with new features.

DevOps also provides enhanced efficiency. This model usually enables teams to take their time and build value. When organizations utilize automated installation and testing, there will not be a need to spend valuable time waiting to install code and even configure machines. Testing and installation can be accomplished by a few clicks on the button when within this self-service portal. Various American financial organizations have adopted DevOps in their business systems, resulting in more than twenty-five percent performance gain when delivering banking updates online (Gregory et al., 2015). DevOps also usually decrease the workloads for the information technology team, such that it can focus on other projects providing the organization with more revenue and benefits.

Further, DevOps can help in ensuring fast recovery and improvement of code performance. When developers release their new products to the market, their next job or project also begins. As such, developers usually lack the justification to avoid or predict potential software issues, and there, the operations team is needed. However, DevOps usually retains software developers involved in upgrading programs or features during a system development life cycle, resulting in better quality code (Lean, 2011). Additionally, a few changes or corrections are necessary since developers can monitor potential code problems while providing fixes. Consequently, since developers usually work with minimized code errors, errors can easily be traced to their sources when a mistake occurs within any software product. With this model's automation of the software development lifecycle, human errors are minimized, enabling organizations to handle changes more quickly (Bell et al., 2006). There are various best practices that help improve the quality of software through implementing DevOps. An organization should not evaluate its progress with an organizational standard. There is also a need to test the current development cycle to identify areas that might need change if one wants to minimize their software development process's time to market (Tessem et al., 2008). For instance, a company can test the process of project execution, which can help identify project restrictions, shortfalls, and limitations. There is a need to set a business standard after these processes' review to illustrate the current software development cycle and changes required to reduce the market time. This model usually helps organizations in seeding their operating routines while boosting their software delivery quality. After analyzing the product delivery process, an organization can see which contributions do not add value to the development process and reduce market speed (Smeds et al., 2015). There is, therefore, a need to remove contribution, not adding value within the software development process. There is a need for open communication between teams and their leaders such that efforts do not go unrecognized. A company can formulate the needed resources and player's structure to ensure efficient and consistent communication between the development and operations teams.

Continuous delivery is another major factor that contributes to time reduction on the market and enhanced software quality. Continuous delivery is usually uninterrupted, and it does not need any intermediate manual interference (Wettinger et al., 2015). Automation usually simplifies and even stimulates communication during deployment and even validates the software development process to ensure regular releases. However, not all organizations usually deploy continuously during their developing software phase. Most organizations usually need to faster for continuous delivery, which delivers new features or even security or software platform for their products. During the software development process, online and medium retailers, for instance, usually tend to deploy.

The automated testing standard is also increased instantly, which helps minimize the time to market with this model. Software testers usually check the programs for their functionality and other variables of every company. However, an organization can enhance automated testing by involving a focused group as well as involving consumers. Entering the general user interface and functionality usually enhances the testing process while also allowing the detection of defects and bugs sooner. For software customers, on the other hand, it has been proved to be highly time-consuming for clients when errors are not corrected or even detected earlier (Wettinger et al., 2015). End to end automated testing usually helps organizations reduce market time as the entire process is accelerated.

CAMS Framework

CAMS is the shorthand for the community, automation, measurement, and sharing, which is a vital DevOps adoption concept. The community, automation, measurement, and sharing framework is beneficial as it usually shows what DevOps entails, that is, individuals, resources, and processes. When applying the DevOps culture into an organization, one quickly learns not to start with software or systems; instead, they should start with people. An organization will facilitate the changes through crucial cultural changes while enabling teams to use available tools more effectively. DevOps comprises repeatable processes, and therefore systems usually continually optimize the transition into production from development (Nielsen et al., 2017). For DevOps, t can rove challenging underestimate DevOps automation's value. DevOps usually simplifies tasks and procedures while doing away with human error and reducing reversals and delays. This automation method usually allows for enhanced results, greater productivity, and even producing value for end-users more effectively and efficiently, which is this model's primary aim. DevOps is a model that allows for quality development even when specific measurements are continually analyzed, measured, and gathered.

Software Quality

A significant and vital factor for business growth and customer satisfaction is quality. A software's quality is usually measured through the software quality characteristics. Software quality characteristics can be defined as a software application product attributes' set by which its quality is evaluated and described. An international standard for software quality evaluation is ISO 9126. This ISO 9126 is a software quality model which identifies six major quality characteristics. These quality characteristics include reliability, efficiency, functionality, portability, maintainability, as well as efficiency. The main characteristics are subdivided into their sub characteristics in total twenty-seven for external and internal quality (Gong et al., 2016). Quality is a vital aspect for customer satisfaction, and organizations need to have a method that aligns with DevOps to enhance quality.

Continuous quality monitoring coupled with continuous testing is majorly tied to continuous deployment, continuous build, and the continuous development process. There is also a need to consider behavior-driven development and test-driven development, which allow for a common shared understanding of the typical application's functionality and operations among all team members, including operations, developers, testers, and end-users. As such, quality is not left for the quality assurance development only, and everyone is required to be responsible for the software's quality. There is a need to consider quality coverage by implementing validation checks and formulating test cases based on the objective rules (Perera et al., 2017). It is also vital to use metrics to manage quality assurance efficiency, such as iteration performance release quality that measures high severity defects, defect leakage, and the number of delivered features.

III. Importance of the research

This research can prove beneficial within the United States for both project managers and organizations across the country. DevOps model offers organizations unique advantages and capabilities that an organization cannot achieve through traditional software development lifecycles. DevOps enhances the collaboration between development and operations teams, leading to substantial cost savings and even higher quality software products. Defects within software product identification and even resolution through DevOps can help find ways of implementing successful and effective testing techniques very quickly. Project management's initial process is usually beneficial for problem resolution. Enhanced productivity usually results in lower development costs and higher software quality. In comparison, low-quality software can advance the current problems, proving time-consuming and even leading to extra costs. Rather than wasting vast amounts of time and effort in solving problems arising within the software, developers with most companies in the United States will focus on creating more successful products. Effective and efficient strategies for improving software quality will help organizations unite to enhance the performance and productivity of their upcoming projects. These approaches aim to help any organization to implement their next project's process with increasing effectiveness and efficiency.

IV. Discussion

A significant success factor for the information technology business is the product or software quality. Quality is vital for customer satisfaction as well. Companies need to implement approaches aligning with DevOps to help enhance software quality. This study's main aim was to explore and identify the relation between DevOps principles and practice and their impact on the quality of software. The conducted literature review provides a broad overview of DevOps and its impact on software quality. Findings from the research indicated that automation is a vital success factor when improving the quality of software. There is also an increasing number of publications that signify DevOps' importance and benefits within the modern and digital world. Information technology solutions usually impact a business' success, and organizations usually find solutions that fulfill the increased demand for custom information technology solutions. In terms of supplying custom solutions quickly into the market, organizations, mostly large and traditional ones, usually struggle to meet customer demands and requirements so of digital products or software. The literature review shows the ever-increasing importance of DevOps for many organizations within the digital transformation process around the world and within the United States.

Traditional software development approaches such as prince II and scrum usually fail since they lack non-agile arrangements. There is a need for organizations to identify teams having automation or capable with necessary skills and need to educate and train them to improve the automation skills. In the case of lacking resources for automation, an organization's management needs to hire employees with the needed skills as action items when implementing and using DevOps. Further, behavior-driven development and test-driven development are some of the best practices organizations follow when practicing DevOps. Behavior-driven development encourages collaborating with all business stakeholders to describe the needed application's functionality. In comparison, test-driven development is a practice that commences with writing quality tests before writing any coding.

Before automation begins, there is a need to formulate a quality environment for automated scenarios. There are several necessary tools such as Cucumber for behavior-driven development, Jenkins for continuous integration, Junit for test-driven development. Quality center, and even EFT for automation. Other tools, including Selenium, JIRA, and even GIT, are used in configuration management. When practicing DevOps, continuous integration is another vital art. Automated scripts require to be finished and added to the continuous integration environment, which results in teams identifying if a build or design is stable and even the environment's status. Further, DevOps are structured within the CAMS model, a recognized model within the studied literature. All relevant DevOps factors are covered by the four dimensions, including culture, automation, measurement, and even sharing.

Another important factor is culture since it changes how teams collaborate and share the responsibility for their system's end-user. Thus, there is a need to initiate a mutual exchange with development and operations teams to break down barriers to collaboration, and each team will learn from the other. However, this collaboration usually relies on employees' working cultures based on their countries and society. Hence company management needs to set up a culture where all employees are conversant with one language. All teams within the organization need to focus on one current issue while also avoiding blame games between the teams. Teams are set up to identify the causes of production defects or issues that affect the software's quality. Also, collaborating when tackling current problems usually enables operation teams, testers, and development teams to become aware of issues facing production and their potential solutions, which can be utilized as mitigation plans or risk future releases. Performance engineers involved in the quality engineering team have enough allowance to concentrate on large-scale tests within an environment similar to those of production. This collaboration usually helps the engineers find scalability, data-driven, and third-party affected performance issues. Collaborations between development and operations usually ensure that organizations can handle tests either within the production environment or even in a staging environment.

For instance, there before, with traditional development methodology, testing teams handled scalability and performance test within their environments. When working within a single environment, teams are required t share the script, framework, and the environment with other teams, which can lead to saving effort and time. As such sharing knowledge and information with other teams is fundamental. Further, the quality engineering team can be divided into several other teams, including a functional one, accessibility team, performance team, and automation team based on individual skill set (Zhu et al., 2016). These teams can share knowledge and information amongst themselves through providing training and needed knowledge. As such, sub-teams can get help from others when they require resources during critical times. Therefore DevOps is a beneficial model that can help organizations in enhancing their software quality. This model requires increased collaboration between the development and operations teams. DevOps within an organization are therefore very beneficial compared to other traditional software development processes.

Conclusion

According to the reviewed literature, there is no sufficient evidence that DevOps helps improve software quality. There is a positive connection between CAMS and software quality which implies that knowledge transfer and exchange can increase software quality. This research's overall discoveries will enable organizations to motivate system designs and DevOps teams to enhance the testing process. The research results suggest that sharing, society, measuring, and automation influence products and services' quality. Hence DevOps can increase software quality if one pays attention to the above details. DevOps is a significant transformation in the information system growth and development. It moderates the distance between end-uses, processes, and developers to facilitate the early detection of issues that may cause problems in the system (Lwakatare et al., 2015). Individuals used the Scrum sprint software reinforced by specifications in the past, but the actual end-user did not have the means to verify these requirements. The case is different with DevOps as it is possible to automate the release and creation of end-user applications periodically.

Additionally, DevOps makes it easy and more effective for operations and developers to work together, which impacts the organizations positively due to improved overall performance. Organizations rely much on software to connect, collaborate, manage sales and maintain market relevance by remaining competitive. Organizations also focus on selling these software applications faster to avoid skipping any possibilities and enhancing consumer demands. Therefore, companies need to employ DevOps methods to save time spent developing software projects in the marketplace and achieving the set goals and objectives.

References

[1] Gill, A.Q., Loumish, A., Riyat, I., and Han, S.,2010, "DevOps for information management systems," VINE Journal of Information and Knowledge Management Systems, vol. 48, no. 1, pp. 122–139.

[2] Nybom, K., Smeds, J., and Porres, I.,2006, “On the impact of mixing responsibilities between devs and ops,” in International Conference on Agile Software Development, pp. 131–143.

[3] Gregory, J., and Crispin, L., 2015, More Agile Testing: Learning Journeys for the Whole Team, Addison-Wesley, Upper Saddle River, N.J.

[4] Laan, S.2011, “IT infrastructure architecture: Infrastructure building blocks and concepts,” U.S.A: Lulu Press.

[5] Bell, T. E., and Thayer, T. A.,2006, "Software requirements: Are they a problem," in Proceedings of the 2nd international conference on Software engineering, pp. 61–68.

[6] Tessem, B., and Iden, J.,2008, “Cooperation between developers and operations in software engineering projects,” in Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering, pp. 105–108.

[7] Smeds, J., Nybom, K., and Porres, I.,2015, “DevOps: a definition and perceived adoption impediments,” in International Conference on Agile Software Development, pp. 166–177.

[8] Wettinger, J., Vasilios, A., and Leymann, F.,2015, "Automated Capturing and Systematic Usage of DevOps Knowledge." Proceedings of the IEEE International Conference on. IEEE Computer Society.

[9] Nielsen, P. A., Winkler, T. J., and Nørbjerg, J.,2017, “Closing the IT Development-Operations Gap: The DevOps Knowledge Sharing Framework.,” in BIR Workshop

[10] Gong, J., Lu, J., & Cai, L. (2016, September). An induction to the development of software quality model standards. In 2016 third international conference on trustworthy systems and their applications (TSA) (pp. 117-122). IEEE.

[11] Perera, P., Silva, R., & Perera, I. (2017, September). Improve software quality through practicing DevOps. In 2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer) (pp. 1-6). IEEE.

[12] Lwakatare, L. E., Kuvaja, P., & Oivo, M. (2015, May). Dimensions of devops. In International conference on agile software development (pp. 212-217). Springer, Cham.

[13] Zhu, L., Bass, L., & Champlin-Scharff, G. (2016). DevOps and its practices. IEEE Software, 33(3), 32-34.

[14] Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). DevOps. Ieee Software, 33(3), 94-100. (Elbert et al., 2016)

[15] Loukides, M. (2012). What is DevOps?. " O'Reilly Media, Inc.". (Loukides, 2012)

[16] Erich, F., Amrit, C., & Daneva, M. (2014). Report: Devops literature review. University of Twente, Tech. Rep. (Erich et al., 2014)

[17] Roche, J. (2013). Adopting DevOps practices in quality assurance. Communications of the ACM, 56(11), 38-43. (Roche, 2013)

[18] Sánchez-Gordón, M., & Colomo-Palacios, R. (2018, October). Characterizing DevOps culture: a systematic literature review. In International Conference on Software Process Improvement and Capability Determination (pp. 3-15). Springer, Cham. (Sánchez et al., 2018)

[19] Lwakatare, L. E., Kuvaja, P., & Oivo, M. (2016). An exploratory study of devops extending the dimensions of devops with practices. ICSEA 2016, 104. (Lwakatare et al., 2016)

[20] Erich, F. M. A., Amrit, C., & Daneva, M. (2017). A qualitative study of DevOps usage in practice. Journal of Software: Evolution and Process, 29(6), e1885. (Erich et al., 2017)