



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Improve Software Quality through practicing DevOps Automation

Sikender Mohsienuddin Mohammad

Sr. DevOps Specialist, Information Technology

419 V Street, Apt D, Sacramento, CA 95818

Abstract— Computer Developers over the past years have implemented new generation programming languages. DevOps is complemented by a range of agile practices designed to improve cooperation between development and operating teams. The main aim of this paper is to investigate how the use of DevOps has affected the quality of software. Another main aim is to explore and identify ways to continuously increase software quality. One way of finding information on this study is to conduct a literature review. The data will provide more information on the current DevOps activities that have been carried out in the software industry. The empirical analysis model has been developed and five theories have been established based on a literature survey. Testing theories using Pearson correlation has achieved research goals. On the basis of the linear regression model where the linear theory is obtained. A literature review was developed to gather quantitative data while DevOps and Quality Assurance experts' interviews were used to determine how DevOps can enhance software quality. Interview reviews, hypothesis testing, and regression tests are recommendations. The quantitative study found that by using CAMS (Culture, Automatic, Measurement, Sharing) systems, software quality is enhanced by practicing DevOps. The biggest

factor in increasing software quality is automation. According to the findings of multiple regression analyses, culture has been shown to boost software quality through integration, evaluation, and sharing. In summary, it is best to use DevOps to generate applications of high quality.

Keywords- Software Quality, DevOps, DevOps Automation, CAMS framework,

I. INTRODUCTION

DevOps's level of growth is one of the main reasons for pushing IT into this modern way of creation and delivery. However, if an organization forgets about effectively accounting for DevOps' quality, half the point is missing. DevOps policies are not only aimed at improving the rate of releases of features, but also at reducing faults. At the end of the day, this is not just about continuous development, but also product enhancement overall [1]. With DevOps' focus on intense automation and development, while agile approaches can result in releases of about six weeks, you can release the code weekly or even daily. This is exciting to innovate frequently for companies with high user demands and/or competitive pressures [2].

Most DevOps new companies are eager to know how long and how much they can save in the development process. The

problem is that speed can occasionally be at the cost of quality and process. Nothing to worry about here for startups. Startups can be loose and quick with little capital and a lack of established or limited customer base. Once any warnings are provided or with broad user repositories, certain bugs reported in development may be patched [3]. That is why big businesses choose low-risk ventures which do not affect business if the program fails as they embark on DevOps [4]. Those also include creative projects, including the development of a new smartphone application or an internal social media platform. The aim of the business is to get fresh ideas, which is ideal for DevOps because it facilitates improvements and retrofits on-the-fly. Expanding DevOps into business-critical areas when we need a fast response and the work of our best team members depends on the user requirements of the program [4]. If DevOps is applied to a sales program that causes users to encounter inactivity or crucial errors for a couple of hours, you can possibly expect the CEO to visit them. It remains a hairy problem about how to guarantee DevOps consistency. Nonetheless, you will have to spend more resources (time and staff) in the quality equation as you begin to scale DevOps across the enterprise.

The creation of DevOps can be called community, motion, theory, or methodology. Depending on many factors that include organizational change and management processes, how you identify the basic elements of DevOps, and how its principles and methods are best used [4]. The good news is that organizations of all sizes and levels of capital have tools and resources to make use of DevOps' technological methods, everywhere they go along with a range of organizational changes. Because DevOps aims to turn software development into a strategic asset, selecting the right tools for the job is important. DevOps encourages speed, performance, and reliability, and these advantages are achieved through proper resources and technology. Information on the technologies used in the software development and operations process is presented to form the "technologies chain" of

technology and services [6]. You can also learn how to use software to implement DevOps methods in different implementation stages and where to find resources for learning how to use the software of DevOps [7]. The aim of this essay is to presents the relationship between the technical practices of DevOps and the improved software quality in terms of instruments and technologies which permit continuous integration and delivery, test automation, and high-speed deployment.

II. LITERATURE REVIEW

DevOps illustrates how the production of software and IT companies will work together in order to cut down the market time. This has achieved such a high profile on the market that the performance of businesses that implement it has been measured.

A. *How does DevOps speed up market time?*

DevOps helps companies to speed up time for marketing their software products [8]. Two of the easiest ways are:

a.) *Fast delivery of modifications and features upgrades*

Once software developers' teams interact with IT operations to submit test applications. When the software product is entirely new, a test environment must be developed for operations. If the program is an improved version of the present product that has been reviewed, operations will need the interfaces and side applications to be modified and implemented [10]. Therefore, if the activity staff tests software, which is a long-term process, a developer can only deploy code in the Application. Nevertheless, from software creation, DevOps removes these time-consuming activities [11]. For instance, automated testing of web app developers provides rapid feedback and automated integration enables changes to the code basis to be made faster. Organizations can, therefore, publish small or incremental software updates while providing new features on the market more quickly.

a.) Improved Efficiency

DevOps allows teams to take longer to build value. If businesses use automated testing and installation, developers don't need to spend time waiting to set up machines or to install code. Both tasks can be achieved through the self-service portal by clicking on a button. In its business systems, several European banks, for instance, have adopted DevOps, resulting in 25% performance gains in delivering online banking updates [12]. It also decreases the workload for the IT team so that they can focus on more demanding projects to give the organization more benefits.

b.) Improvement of Performance code and fast recovery

When developers release products on the market, their next project is also begun the job. Therefore, they do not typically have justification to predict or avoid potential software problems; it is all up to the operating team. Yet DevOps retains developers involved in upgrading every feature or program during the software's life cycle, which results in higher quality code [13]. Furthermore, a few corrections are needed because developers can monitor possible code problems and fix them. As developers work with lesser pieces of code, bugs can be easily traced to their resources if a mistake happens in any product. Therefore, human mistakes are minimized due to DevOps' automation of the software life cycle, enabling businesses to make changes more quickly [14].

B. Best practices to improve software quality using DevOps

An organization's progress could never be evaluated without a company standard. You may also test the current evolution cycle to find out which areas require change if you want to minimize time to markets for your software development process [15]. You may, for example, test how a project should be executed and recognize any project limitations, shortfalls, and restrictions. One should set a business standard after reviewing these processes to demonstrate the existing product development cycle and changes

required to minimize market time. DevOps helps you to speed up the operating routine and boost the quality of your software delivery.

One can see contributions that do not add value to the process and reduce speed when you analyze your software delivery process [17]. It is important to remove these contributions from the development process, but ensure that open contact between you and your team is available so that efforts are not wasted. The right player's structure and resources to ensure consistent and efficient communication between the teams can be created. For instance, you can organize a group that needs business owners, right analytics, and DevOps tools, more importantly.

Continuous delivery is one of the key factors that have contributed to time reduction on the market. It is a continuous pipeline that does not require any intermediate manual interference [18]. Automation stimulates and simplifies communication during ongoing deployment and validates the software process in order to ensure prompt releases. Nonetheless, not all organizations are continually deployed in their evolving software phase. Typically, companies need to move more rapidly for continuous delivery in order to deliver new features or software or security platforms for their product [18]. Throughout their software creation process, medium and online retailers, for example, tend to deploy.

The standard of automated testing is increased instantly to reduce the time to market with DevOps. Testers check the program for its functionality and other variables of every organization. However, this research phase can be improved thanks to the involvement of consumers and a focused public. Entering the overall user interface and functionality would not only boost the testing process but also allow you to detect bugs and defects sooner. It has been shown to be highly time-consuming for customers for the software system when bugs are not detected or corrected at the early stages [18]. A good example is the consideration of cross-browser compatibility testing to

check the test automation. The entire process is accelerated with end-to-end automated testing, which decreases market time.

C. CAMS Framework

The CAMS framework is very effective since it illustrates what DevOps really is about — individuals, processes, and resources — and it highlights the value of creating and maintaining all three during a DevOps journey. CAMS (short for the community, automation, measurement, and sharing) is a foundational concept of DevOps adoption. One will quickly discover out, when they are looking at how DevOps can be applied, that you should not start with software or systems-start with people, with DevOps culture embedded into a business. A crucial cultural shift is a way companies will facilitate process change and enable teams to use DevOps tools more effectively. DevOps is all "repeatable" processes — systems that continually optimize the transition from development into production[20]. It is impossible to underestimate the value of automation for DevOps. It simplifies procedures and tasks, eliminates human error, and reduces delays and reverse reversals. The automation method allows for greater productivity, improved results, and produces value for the end-user more efficiently, which is the aim of DevOps in the first place. DevOps enables quality development even when specific measurements are continually gathered, measured, and analyzed.

DevOps experience	Period
24%	Less than 1 year
62%	1 to 3 years
14%	More than 3 years

Table: This table shows the experience of developers in using DevOps to improve the quality of their software.

Professional track	Proportion
Project management	11%
operation	23%
Development	21%
Quality assurance	45%

Table 2: Table showing how effective DevOps is in various professional tracks.

III. RESEARCH MODEL

The hypotheses in this research study involved the verification of relationship variables which are DevOps and software quality.

Null Hypothesis, H₀: There are no relations between DevOps and the improved quality of software.

The alternative hypothesis, H_a: There is a relationship between DevOps application and the quality of software.

IV. IMPORTANCE OF RESEARCH TO THE UNITED STATES

This research will benefit the United States in improving the quality of software that is utilized in many businesses and project management. The identification and resolution of defects using DevOps will assist in seeking ways to incorporate successful testing techniques as quickly as possible. The initial process of project management is a win-win situation for the resolution of problems. Greater productivity leads to higher software quality and lower costs [20]. On the other hand, poor quality of software aggravates the problems and could become time-consuming and costly. Developers in many companies in the U.S will concentrate on creating a successful product, instead of wasting long stretches of time solving software problems. effective strategies for enhancing software quality will help businesses improve productivity and performance for your next project. Such approaches are meant to assist any companies so that they can perform the process of their next project with increasing efficiency.

V. CONCLUSION

Based on the literature review, there is enough evidence that shows that the software quality is improved when DevOps is applied. There is also a positive relationship in terms of software quality and CAMS (software culture, automation, measurement, and sharing). This means that the exchange of knowledge can improve software quality. The findings of this research will allow companies to conduct DevOps and systems design teams to agree to enhance testing. Study results have clearly shown that the quality of goods is influenced by society, automation, measuring, and sharing. Therefore, if you take proper account of the above details, DevOps practice should boost software quality. DevOps is a big shift in the growth of the information system [21]. DevOps reduces the distance between developers, processes, and end-users so that issues can be found sooner. In the past Scrum, sprint software was supported by specifications, but the real end-user did not verify such requirements. With DevOps, we can periodically automate the creation and release of end-user applications. DevOps also makes working together more efficiently and effectively between developers and operations [22]. Today, companies depend heavily on software to work together, connect, compete, and manage sales more proficiently. We also aim to sell these applications quicker, so they can quickly adapt to consumer demands and not skip any possibilities. To order to achieve these goals companies must implement DevOps methods to reduce time to the marketplace to their software development projects.

REFERENCES

- [1] Sikender Mohsienuddin Mohammad, "DEVOPS AUTOMATION AND AGILE METHODOLOGY ", International Journal of Creative Research Thoughts (IJCRT), ISSN:2320-2882, Volume.5, Issue 3, pp.946-949, August-2017, Available at :<http://www.ijcrt.org/papers/IJCRT1133441.pdf>
- [2] Tarvo, A.,2009, "Mining Software History to Improve Software Maintenance Quality: A Case Study", *IEEE Software*, vol. 26, no. 1, pp. 34-40. Available: 10.1109/ms.2009.15.Sacks, M., 2012, Pro website development and operations: Streamlining DevOps for large-scale websites, Apress, [New York].
- [3] Le Goues, C., and Weimer, W.,2012, "Measuring Code Quality to Improve Specification Mining", *IEEE Transactions on Software Engineering*, vol. 38, no. 1, pp. 175-190, 2012. Available: 10.1109/tse.2011.5.
- [4] Virmani, M.,2015, "Understanding DevOps & bridging the gap from continuous integration to continuous delivery. *Fifth International Conference on Innovative Computing Technology (INTECH 2015)*. DOI:10.1109/intech.2015.7173368
- [5] Riungu-Kalliosaari, L., Mäkinen, S., Lwakatare L. E., Tiihonen, J. and Männistö, T.,2009, "DevOps Adoption Benefits and Challenges.
- [6] Humble, J. and Farley, D., 2010, "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (Adobe Reader). Pearson Education.
- [7] Limoncelli, T., Chalup, S., Hogan, C., and Limoncelli, T., 2014, The practice of cloud system administration.
- [8] Hüttermann, M., 2012, "Building Blocks of DevOps. DevOps for Developers," 33-47. DOI:10.1007/978-1-4302-4570-4_3.
- [9] Kim, G., Love, P., and Spafford, G., 2014, Visible Ops Security: achieving common security and IT operations objectives in 4 practical steps, Eugene, OR: IT Process Institute.
- [10] 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.
- [11] 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.
- [12] Gregory, J., and Crispin, L., 2015, More Agile Testing: Learning Journeys for the Whole Team, Addison-Wesley, Upper Saddle River, N.J.
- [13] Laan, S.2011, "IT infrastructure architecture: Infrastructure building blocks and concepts," U.S.A: Lulu Press.
- [14] 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.

- [15] 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.
- [16] Huttermann, M., 2012, "DevOps for developers. Apress.
- [17] 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.
- [18] Wettinger, J., Vasiliou, A., and Leymann, F.,2015, "Automated Capturing and Systematic Usage of DevOps Knowledge." Proceedings of the IEEE International Conference on. IEEE Computer Society.
- [19] Hneif, M., and Lee, S.P.,2011, "Using Guidelines to Improve Quality in Software Nonfunctional Attributes", *IEEE Software*, vol. 28, no. 6, pp. 72-77. Available: 10.1109/ms.2010.157.
- [20] Nielsen, P. A., Winkler, T. J., and Nørbjerg, J.,2017, "Closing the IT Development-Operations Gap: The DevOps Knowledge Sharing Framework.," in BIR Workshops.
- [21] Herring, M.,2015, "Continuous everything in DevOps," Accenture. Retrieved from: <https://www.accenture.com/us-en/blogs/software-engineering-blog/herring-continuous-everything-in-devops>
- [22] Boehm, B.,2006, "A view of 20th and 21st-century software engineering," in Proceedings of the 28th international conference on software engineering, pp. 12–2

