IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 7, July 2023

DOI: 10.17148/IJARCCE.2023.12727

Survey Paper on Improve Software Quality through Practicing DevOps

Ms Kirti Ankalgi[1], Ms Nikita Ghatage[2], Dr. Pijush Barthakur[3]

Student of KLS Gogte Institute of Technology, Belagavi[1,2]

Assistant Professor of KLS Gogte Institute of Technology, Belagavi[3]

Abstract: DevOps, an extension of certain agile practices, combines various patterns to enhance collaboration between development and operations teams. The primary objective of this research is to investigate the impact of DevOps implementation onsoftware quality, while also exploring effective strategiesto improve quality efficiently. Through an extensive literature survey, current DevOps practices in the industry were explored, leading to the development of a conceptual research model with five derived hypotheses.The research objectives were achieved by conducting hypothesis testing using Pearson correlation and deriving a linear model based on linear regression analysis. To gather quantitative data, an online questionnaire was employed, supplemented byinterviews with DevOps and Quality Assurance experts to identify how DevOps can enhance software quality.

The feedback from interviews, along with hypotheses testing using regression analysis, served as the basis for recommendations. The findings from the quantitative study reveal that software quality experiences significant improvement through the adoption of DevOps, particularly when following the CAMS (Culture, Automation, Measurement, Sharing) framework. Among the factors, automation emerged as the most critical element for enhancing software quality. The results of multiple regression analysis further emphasizethe importance of culture, automation, measurement, and sharing in the pursuit of improved software quality.In conclusion, this study strongly advocates for the implementation of DevOps to attain high-quality software. By incorporating the CAMS Framework and adhering to principles such as automation, organizationscan effectively elevate the overall quality of their software products.

Keywords: DevOps, CAMS Framework, Quality, ISO 9126, Automation"

I. INTRODUCTION

Over the past two decades, software has evolved into an indispensable component of businesses. Softwareengineering researchers have continuously introduced innovative languages, software architectures, development tools, and technologies such as Cloud Computing, Crowdsourcing, Application Programming Interfaces (APIs), Rest services, big data, and Internet of Things (IoT).

In the IT industry, software development process models serve as a common framework to structure, plan, and controlthe efficient and productive development of information systems. Various software development life cycle models have been defined and designed for this purpose. Eachmodel follows a unique series of steps tailored to ensure success in software development. Some traditional models include the Waterfall Model, Iterative Model, Spiral Model,V-Model, and Big-Bang Model. In 2001, the Agile methodology was introduced, which led to the adoption of concepts like Scrum and Kanban. More recently, DevOps emerged as an improved version of Agile, with a focus on operational aspects.

DevOps promotes seamless collaboration and communication between developers and operations teams todeliver software and services rapidly, reliably, and with highquality. The term 'DevOps' stems from combining "Dev" from Developers and "Ops" from Operations. It empowers teams with shared responsibilities and accountability throughout the service's lifecycle, from development to deployment and support. Cross-functionality, shared responsibilities, and trust are emphasized in a DevOps environment, extending the continuous development goals of Agile to continuous integration and release. Automation of change, configuration, and release processes is encouraged in order to accommodate frequent releases.

This research aims to investigate whether the quality of software improves when implementing DevOps in software companies. The analysis identifies key factors that need to be considered for enhancing software quality and how companies can effectively practice DevOps.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 7, July 2023

DOI: 10.17148/IJARCCE.2023.12727

II. BACKGROUND AND RELATED WORK

A. SDLC

The Software Development Life Cycle (SDLC) is a conceptual model that outlines the steps involved in planning, designing, testing, and deploying software. Various methodologies are employed in the industry toeffectively manage the SDLC. One notable methodology is Agile, which was developed as a response to the limitationsof rigid plan-driven process models like Waterfall and Rational Unified Process (RUP), among others. Agile was also introduced to address the challenges faced by other resistant methodologies.

Agile is an innovative approach to project management, primarily adopted in software development projects. It emphasizes iterative and incremental development, promoting flexibility and adaptability in the development process. Unlike traditional plan-driven models, Agileencourages collaboration and continuous customer feedback, leading to better outcomes and customer satisfaction.

By embracing Agile, software development teams can respond quickly to changing requirements, deliver valuable features incrementally, and maintain a focus on delivering high-quality software. This dynamic and customer-centric approach has made Agile a preferred methodology in the software industry, facilitating the development of successful software products.

The software development market has experienced a remarkable surge marked by unpredictability and rapid changes. These changes are primarily driven by evolving client requirements and the positive adaptation to change requests. To meet these dynamic demands and ensure highercustomer satisfaction, many companies have embraced agiledevelopment practices, allowing them to achieve frequent releases and responsiveness.

While the majority of companies have adopted agile practices, some have taken a step further by implementing specific agile practices to alleviate bottlenecks in their existing processes. Notable tech giants such as IBM, Facebook, and Microsoft have ventured into Continuous Deployment as a means to streamline their development efforts. Continuous Deployment, as defined by Claps, Svensson, & Aurum, is a challenging endeavor due to its significant impact on system stability. Nevertheless, it opensup new business opportunities while introducing novel challenges for software companies. Implementing Continuous Deployment is not without its hurdles, as it requires careful consideration of system stability and robust testing mechanisms. However, the rewards of faster and more streamlined development cycles,along with improved customer satisfaction, have incentivized these companies to explore and adopt this approach.

B. DevOps

"DevOps is a collection of principles aimed at enhancing collaboration between development and operations teams. It seeks to address shared goals, incentives, processes, and tools to bridge the gaps that may naturally exist among different groups. While achieving complete alignment of shared goals and incentives may not always be feasible due to inherent conflicts, they should at least be harmonized witheach other."

The primary objective of DevOps is to identify and eliminatethe divisions between Development and Operations teams. Professionals involved in software development often approach their tasks with a focus on delivering product changes. The key goals of DevOps, as identified in, are as follows:

- Deliver measurable business value throughcontinuous and high-quality service delivery.
- Prioritize simplicity and agility in all aspects, including technology, processes, and human factors.
- Remove barriers between development and operations by fostering trust, shared ownership, anda culture of innovation and collaboration.
- Manage dynamic compliance, considering the evolving laws and regulations related to access anddata sharing.

The Phoenix Project has described the division that can be bridged when practicing DevOps, including Security Compliance and IT, Development and Quality Assurance, Marketing and IT, and Business Goals and IT Delivery.

Behr and Kim of The Phoenix Project emphasize that DevOps does not guarantee flawless outcomes with frequent deliveries, stable platforms, and perfect business-IT alignment. Instead, the focus should be on creating a collaborative environment where everyone can contribute their best to resolve issues and support the business.

DOI: 10.17148/IJARCCE.2023.12727

According to the State of DevOps Report 2016, implementing DevOps has led to improved organizational performance, revenues, and profitability. The adoption of DevOps practices has been increasing, with the number of DevOps teams rising from 16% in 2014 to 19% in 2015 and 22% in 2016. High-performing companies have significantly increased their deployment frequency, leading to faster lead times, outperforming low-performing companies by deploying changes 200 times more frequentlyand with 2,555 times faster lead times. Notably, large companies like Amazon and Netflix have demonstrated the effectiveness of DevOps by deploying changes thousands oftimes per day. These compelling facts illustrate why companies have been increasingly embracing DevOps as a crucial approach in their software development processes.

C. CAMS Framework

The CAMS Model was devised by DevOps pioneers John Willis and Damon Edwards, representing the core values of Culture, Automation, Measurement, and Sharing. This model has been widely embraced by DevOps practitioners as the fundamental set of principles for successful DevOps implementation.

Culture assumes paramount importance as the foundation for any DevOps implementation. It emphasizes the need for teams to work together, sharing responsibilities for the end-users of their applications. By fostering a culture of collaboration, DevOps not only encourages the adoption of agile practices in operations but also facilitates a knowledge exchange between developers and operations teams, effectively breaking down traditional barriers.

Automation plays a crucial role in enabling the adoption of DevOps. The right combination of people, processes, and tools is essential to establish an automation framework for DevOps practices. Operations and testing teams often possess valuable insights into application performance, and they should educate developers about the significance of maintaining application performance in large-scale environments under heavy loads. By providing automated mechanisms to monitor performance across all environments, from continuous integration and testing to production deployment, a shared language of performance can be established.

Measurements, including business key performance indicators, system metrics, and application behaviour, are integral to a successful DevOps model. These measurements should be open, transparent, accessible, meaningful, and easily visualized by all members of the DevOps team. By addressing performance aspects in early testing stages, performance engineers on testing teams can focus on conducting large-scale load tests in production-like environments, ensuring robust performance.

Sharing tools, discoveries, and lessons learned is critical to the success of any organization's DevOps implementation [7]. Collaboration thrives on the exchange of ideas and problem-solving, and this is at the heart of the DevOps philosophy. Emphasizing openness and transparency, DevOps fosters an environment where sharing knowledge and insights is highly valued.

D. Software Quality

Quality stands as the utmost critical factor in ensuring customer satisfaction and driving business growth. Evaluating the quality of software involves assessing various software quality characteristics, which represent a set of attributes describing and gauging the software product's overall quality. To facilitate the evaluation process, the International Standard ISO 9126 serves as a benchmark for software quality assessment.

The ISO 9126 software quality model outlines six primary quality characteristics: Functionality, Reliability, Usability, Efficiency, Maintainability, and Portability. These key characteristics are further divided into twenty-seven sub-characteristics, encompassing both internal and external aspects of software quality.

Customer satisfaction heavily relies on the quality of the products and services offered by companies. Hence, organizations must adopt a DevOps approach to elevate thesoftware quality continually. The principles of continuous testing and continuous quality monitoring are intrinsically connected to the concepts of continuous development, continuous build, and continuous deployment. Emphasizingthese aspects within the DevOps framework ensures that software quality is a consistent focus throughout the software development and delivery process, ultimately contributing to customer satisfaction and business success. Customer satisfaction heavily relies on the quality of the products and services offered by companies. Hence, organizations must adopt a DevOps approach to elevate the software quality continually. The principles of continuous testing and continuous quality monitoring are intrinsically connected to the concepts of continuous development, continuous build, and continuous deployment. Emphasizingthese aspects within the

DOI: 10.17148/IJARCCE.2023.12727

DevOps framework ensures that software quality is a consistent focus throughout the software development and delivery process, ultimately contributing to customer satisfaction and business success. Test Driven Development (TDD) and Behaviour Driven Development (BDD) play crucial roles in establishing a shared understanding of the expected operations and functionalities of an application among all team members, including users, developers, testers, and operations personnel. The responsibility for maintaining software quality should not rest solely on the shoulders of the Quality Assurance (QA) team; rather, it is a collective responsibility shared by everyone involved in the development process. To ensure comprehensive quality coverage, validation checks and test cases should be meticulously designed based on well-defined objective rules. Implementing metrics is essential to manage QA efficiency effectively. Key metrics, such as iteration performance and release quality, which includes measurements like defect leakage, number of high severity defects, and the number of features delivered, can help gauge the success of DevOps practices.

The research paper titled "Evaluating the Impact of DevOps Practice in Sri Lankan Software Development" highlights that companies in Sri Lanka have successfully embraced the DevOps environment. As a result, there has been a notable improvement in responsiveness to business needs, adaptability to new technologies, and enhanced process quality since the adoption of DevOps practices.

III. RESEARCH MODEL

Based on a comprehensive literature review and the main research objective, the researchers have devised a conceptual framework illustrated in Figure 1. The framework is divided into two sections, encompassing essential factors for DevOps implementation and its impact on software quality.

The formulated hypotheses in this research aim to examine the relationships between variables, aligned with the primary and secondary objectives. The following hypotheses have been derived:

Hypothesis 1: Implementing DevOps leads to an improvement in the Quality of the Software.

H10: There is no significant relationship between DevOps implementation and the Quality of the software.

H1A: There is a significant relationship between DevOps implementation and the Quality of the software.

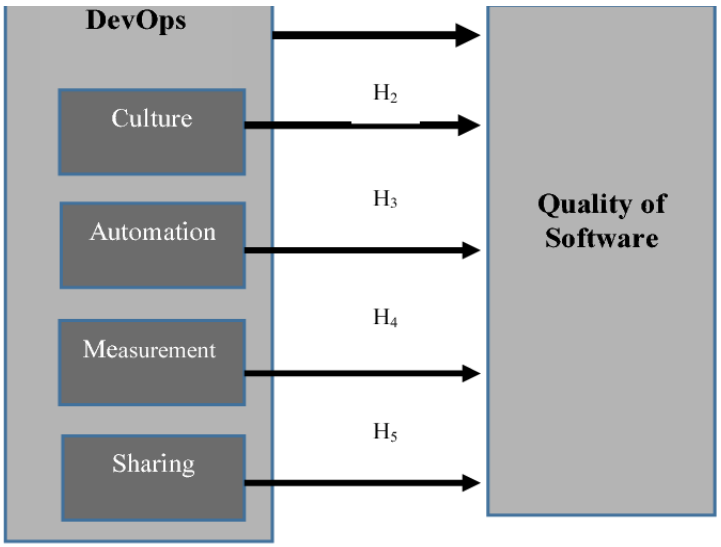

Fig. 1. Research Model and research hypotheses [1]

Hypothesis 2: The Culture of DevOps positively influences the Quality of the Software.

H20: There is no significant relationship between the Culture of DevOps and the Quality of the software.

H2A: There is a significant relationship between the Culture of DevOps and the Quality of the software.

Hypothesis 3: Automation in DevOps contributes to the improvement of the Quality of the Software.

DOI: 10.17148/IJARCCE.2023.12727

H30: There is no significant relationship between Automation in DevOps and the Quality of the software. H3A: There is a significant relationship between Automation in DevOps and the Quality of the software.

Hypothesis 4: Sharing in DevOps enhances the Quality of the Software.

H40: There is no significant relationship between Sharing in DevOps and the Quality of the software.

H4A: There is a significant relationship between Sharing inDevOps and the Quality of the software.

Hypothesis 5: Measurement in DevOps has a positive impacton the Quality of the Software.

H50: There is no significant relationship between Measurement in DevOps and the Quality of the software. H5A: There is a significant relationship between Measurement in DevOps and the Quality of the software.

To validate these hypotheses and measure the variables"Improve software quality through practicing DevOps" and "Quality of the software," a carefully designed online questionnaire was distributed among software professionalsworking in companies that practice DevOps. The questionnaire included five-point Likert scale questions and open-ended questions to gather detailed insights. Additionally, face-to-face interviews were conducted with DevOps experts to gain further information.

The independent variable, "Practice DevOps," was measured through the dimensions of Culture, Automation, Measurement, and Sharing. The dependent variable of the study was the "Quality of the software." Indicators for each variable were identified based on existing literature sources,as depicted in Table 1.

TABLE1: OPERATIONALIZATION TABLE

Variables	Indicators	Source
Software Quality	1. Functionality	[9]
	2. Reliability	
	3. Efficiency	
	4. Maintainability	
	5. Usability	
	6. Portability	
Practice DevOps	1.Culture	[7]
	2. Automation	
	3. Measurement	
	4. Sharing	

Upon preparing the questionnaire, the researchers conducteda pilot study involving 15 respondents. To ensure the internal consistency of the measured variables, the statisticalmethod Cronbach's Alpha was utilized, selecting questions with a Cronbach's alpha above 0.7.

In Sri Lanka, there are approximately 200 registered Software development organizations, classified into three scales: small, medium, and large companies. The selection of respondents was based on their DevOps experience, specifically those who had prior experience with practicing DevOps.

For data analysis, the researchers aimed to assess the qualityof the data and validate the developed hypotheses. Graphical statistical tools such as the normality curve and boxplot diagrams were employed to test the normal distribution of the data. Once it was confirmed that all variables exhibiteda normal distribution, Pearson correlation was used to infer the relationships between the independent and dependent variables.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 7, July 2023

DOI: 10.17148/IJARCCE.2023.12727

IV. RESULTS AND ANALYSIS

A. Quantitative Data Analysis and HypothesesTesting:

An online questionnaire was distributed among more than 300 software professionals working in companies that practice DevOps and have prior DevOps experience. Out ofthe responses received, 120 were selected for further analysis, based on whether respondents had previous DevOps experience and their professional background.

Figure 2 illustrates the DevOps experience of the respondents. The majority, 62%, have an experience of 2 to3 years with DevOps. About 24% have less than 1 year of DevOps experience, while 14% are professionals who have been practicing DevOps for more than 3 years.

Figure 3 presents the development methodology used by respondents in their respective companies. The largest portion, 66%, comes from companies that practice both DevOps and Agile methodologies

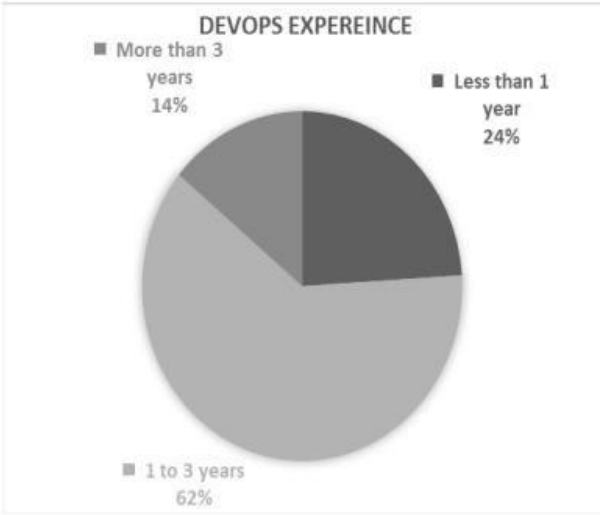

Fig. 2. DevOps Experience

Fig. 3. Current Development Methodology

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 7, July 2023

DOI: 10.17148/IJARCCE.2023.12727

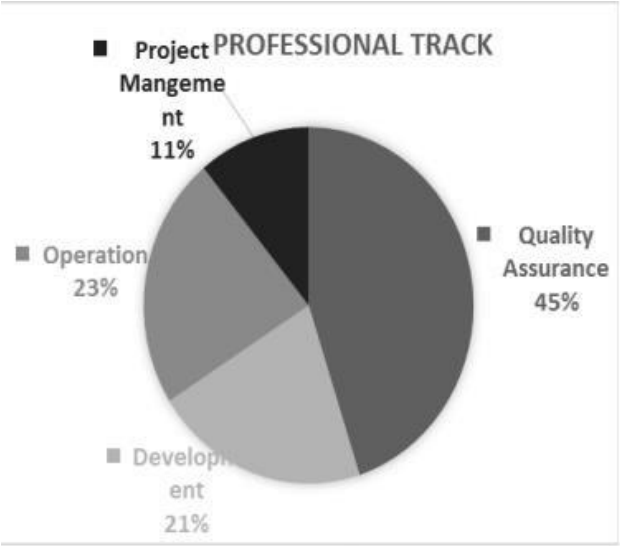

Fig. 4. Professional Track

In terms of the software development methodology, 24% of the respondents come from companies that practice Agile. Interestingly, 10% of the respondents have prior DevOps knowledge despite working in companies that follow the traditional waterfall methodology.

Figure 4 depicts the distribution of respondents based on their designations or tracks. The majority, 45%, belong to the Quality Assurance track. 21% are from the Developmenttrack, 23% from Operations, and the remaining 11% are from project management.

Table 2 provides a summary of the correlation analysis. Therelationships between the dependent and independent variables exhibit strong positive correlations at a 99% confidence level. Based on these results, all five hypotheseshave been confirmed, showing a significant positiverelationship between the dependent and independent variables.

TABLE 2: PEARSON CORRELATION

		Quality of Software	Relationship
Practice DevOps	Pearson correlation	0.76**	Strong
	Sig. (2-tailed)	0	
Culture	Pearson correlation	0.741**	Strong
	Sig. (2-tailed)	0	
Automation	Pearson correlation	0.758**	Strong
	Sig. (2-tailed)	0	
Measurement	Pearson correlation	0.704**	Strong
	Sig. (2-tailed)	0	
Sharing	Pearson correlation	0.717**	Strong
	Sig. (2-tailed)	0	
**Correlation is significant at the 0.01 level (2-tailed)			

DOI: 10.17148/IJARCCE.2023.12727

The fundamental success factor in the IT business lies in thequality of the software or product. The primary objective of this research was to explore and establish the relationship between DevOps practices and their impact on achieving software quality. To achieve this goal, a multiple regressionanalysis was conducted to formulate a model that accurately represents the relationship between DevOps and Quality. The equation is represented using the following notation:

Quality = β0 + β1C + β2A + β3M + β4SWhere:

Quality is the dependent variable, symbolizing the overallsoftware quality.

C represents Culture, an independent variable reflecting theinfluence of DevOps culture on software quality.

A stand for Automation, an independent variable indicatingthe impact of DevOps automation practices on software quality.

M denotes Measurement, an independent variable representing the effect of DevOps measurement practices onsoftware quality.

S represents Sharing, an independent variable indicating theimpact of DevOps sharing practices on software quality.

By utilizing this multiple regression analysis and the resulting equation, the research aims to shed light on the relationship between DevOps practices and software quality,thereby assisting IT businesses in understanding the vital role of DevOps in achieving higher levels of software excellence.

$$\text{Software Quality} = 1.409 + 0.176(C) + 0.227(A) + 0.096(M) + 0.172(S)$$

B. Recommendation

The primary objectives of this research are focused on identifying effective strategies to enhance software quality in a DevOps working environment. To achieve this, qualitative questions and interviews were conducted with DevOps and Quality Assurance experts, considering factorslike Culture, Automation, Measurement, and Sharing.

Quality remains a critical aspect for customer satisfaction, and adopting a DevOps approach is essential to improve software quality. The research findings emphasize the significance of Automation as a crucial success factor for quality improvement. However, it is essential for software teams to carefully evaluate the Return on Investment (ROI)before proceeding with automation to ensure sustainability in the long run.

To implement DevOps successfully, companies need to identify or develop a team with automation and technical skills and provide appropriate training to enhance their automation capabilities. In situations where automation resources are scarce, recruitment of employees with the required skills becomes a crucial action item.

Best practices in DevOps involve Test Driven Development (TDD), Behaviour Driven Development (BDD), and Acceptance Test Driven Development (ATDD). TDD startswith writing tests before coding, BDD involves working with business stakeholders to describe desired functionalityin natural language, and ATDD focuses on finding scenariosfrom an end-user perspective.

Before implementing automation, establishing a quality environment is vital. Various tools, such as Jenkins for Continuous Integration, Cucumber for BDD, Junit for TDD,GIT for configuration management, and others like Quality Centre, JIRA, ALM for Test lifecycle and defect management, Selenium, QTP, and UFT for automation, canbe used effectively in the DevOps context.

Continuous Integration (CI) plays a significant role in DevOps practices. Automated scripts should be completed and integrated into the CI environment to identify the stability and status of the build and environment. Culture is a crucial factor as it influences the collaboration and responsibility-sharing among teams. Encouraging mutual exchange between Development and Operations teams helps break down walls and fosters a culture of learning from each other. Management should create an environment where all employees speak a common languageand work together to address current problems, avoiding finger-pointing and identifying root causes.

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 7, July 2023
DOI: 10.17148/IJARCCE.2023.12727

Performance engineers should focus on large-scale load testsin production-like environments with close collaboration with Operations. Sharing environments, frameworks, and scripts across teams can save time and effort.

In conclusion, adopting DevOps practices can significantly improve software quality, but it requires a focus on factors such as Automation, Collaboration, and a culture of shared responsibility. Proper training, tool selection, and continuous integration are also critical components for success.

V. CONCLUSIONS

The primary objective of this paper is to conduct an analysisto determine whether the implementation of DevOps leads to an improvement in software quality. To achieve this, datawas gathered through online questionnaires and interviews with DevOps experts in the software development industry.

After collecting and analysing the data, the research revealeda positive relationship between the practice of DevOps and the quality of software. It was observed that software qualitytends to increase when DevOps is implemented. Furthermore, there was a strong positive correlation betweenCulture, Automation, Measurement, and Sharing with software quality. This implies that practicing automation andsharing knowledge positively influences software quality.

The findings of this research provide valuable insights for companies that practice DevOps and quality engineering teams, enabling them to make informed decisions to enhancetheir testing practices. The research results clearly indicate that factors such as culture, automation, measurement, and sharing have a significant impact on the quality of products. Therefore, by correctly considering these factors and practicing DevOps, companies can improve software quality.

REFERENCES

[1] Agile Manifesto, "Agile Manifesto," Agile Manifesto, 2017. [Online]. Available: http://www.agilemanifesto.org/.. [Accessed 10 04 2017].
[2] A. Aurum, R. B. Svensson, and G. Claps, "On the journey to continuous deployment: Technical and social challenges along the way," Information and Software Technology, vol. 57, pp. 21-31, 2015.
[3] M. Huttermann, "Beginning DevOps for Developers," inDevOps for Developers, 2012, pp. 3-13.
[4] D. L. Farroha and B. S. Farroha, "A Framework for Managing Mission Needs, Compliance, and Trust in the DevOps Environment," In Military Communications Conference, pp. 288-293, 2014.
[5] K. Behr, G. Spafford, and G. Kim, The Phoenix Project:A Novel About IT, DevOps, and Helping Your Business Win, US: IT Revolution Press, 2013.
[6] Puppet and Dora, "2016 State of DevOps," 2016.
[7] D. Farley and J. Humble, Continuous Delivery: ReliableSoftware Releases through Build, Test, and Deployment Automation, Pearson Education, 2010.
[8] M, "diazhilterscheid," 28 06 2016. [Online]. Available: zhilterscheid.de/en/ready-for-cams-the-foundation-of-devops/. 2017].
[9] ISO, "ISO/IEC 9126-1:2001 - Software engineering – Product quality -- Part 1: Quality model," 06 2001. [Online]. Available: https://www.iso.org/standard/22749.html.[Accessed 10 03 2017].
[10] Capemgini and Sogeti, "DevOps with Quality," 2016. [Online]. Available: https://www.sogeti.com/globalassets/global/dow nloads/testing/pov_devops-with-quality_ok.pdf. Accessed 20 01 2017]