Improve software quality through practicing DevOps

Conference Paper · September 2017

DOI: 10.1109/ICTER.2017.8257807

3 authors, including:

Indika Perera

172 PUBLICATIONS **741** CITATIONS

SEE PROFILE

2017 International Conference on Advances in ICT for Emerging Regions (ICTer): 013 - 018

Improve Software Quality through Practicing DevOps

Pulasthi Perera, Roshali Silva, Indika Perera

Department of Computer Science & Engineering, University of Moratuwa
Sri Lanka
pulasthioshan@gmail.com
roshalisilva@gmail.com
indika@cse.mrt.ac.lk

***Abstract*— DevOps is extended from certain agile practices with a mix of patterns intended to improve collaboration between development and operation teams. The main purpose of this paper is to conduct a study on how DevOps practice has impacted to software quality. The secondary objective is to find how to improve quality efficiently. A literature survey has carried out to explore about current DevOps practices in industry. According to the literature survey, the conceptual research model was developed and five hypotheses were derived. Research objectives were accomplished by testing hypotheses using Pearson correlation. A linear model is derived based on the linear regression analysis. An online questionnaire was used to collect quantitative data whereas interviews with experts on DevOps and Quality assurance have been used to identify how to improve the quality of software by practicing DevOps. Recommendations are given based on interview feedback, hypotheses testing with regression analysis. According to the quantitative study, researchers have identified that quality of the software gets improved when practice DevOps by following CAMS (Culture, Automation, Measurement, Sharing) framework. Automation is the most critical factor to improve the software quality. As per the results of multiple regression analysis, it has proved culture, automation, measurement and sharing are important factors to consider to improve quality of the software. In conclusion it can be recommended to use DevOps to achieve high quality software.**

***Keywords*— DevOps, CAMS Framework, Quality, ISO 9126, Automation**

I. INTRODUCTION

Software has become an essential part of the business over the last two decades. Software engineering researchers introduce new generation of languages, software architectures, development tools and technologies such as Cloud Computing, Crowdsourcing, and Application Programming Interfaces and Rest services, Big data, Internet of Things.

The software development process models used in the IT industry follows a common framework used to structure, plan and control the process of developing an information system in an efficient and productive way. There are number of software development life cycle models defined and designed to be practiced during the software development process. Each process model follows a series of steps unique to its type, ensuring success in the process of software development. Traditional software development life cycle models are Waterfall Model, Iterative Model, Spiral Model, V-Model, Big-Bang Model. In 2001 Agile was introduced to the industry and companies used to practice agile concepts such as Scrum, Kanban. Few years ago industry introduced DevOps which was an improved version of Agile focusing on operational aspects.

DevOps is a set of methods in which developers and operations communicate and collaborate to deliver software and services rapidly, reliably and with higher quality. The word 'DevOps' has come up with Dev from Developers and Ops from Operation. DevOps is sharing of tasks and responsibilities within a team empowered with full accountability of their service and its underlying technology stack; from development, to deployment and support.

In a DevOps environment, cross functionality, shared responsibilities and trust are promoted. DevOps essentially extends the continuous development goals of the agile movement to continuous integration and release. In order to accommodate continuous releases, DevOps encourages automation of the change, configuration and release processes.

The objective of this research is to study whether quality of the software gets improved when practicing DevOps in software companies. This analysis has indicated the factors need to be considered to improve the quality of the software and how companies can practice DevOps

The organization of the paper is as follows: Section 2 discusses about DevOps concepts, CAMS framework and measuring quality of the software. Section 3 has introduced the methodology carried out for the research and the research model. Section 4 has presented the data analysis and output of the results with recommendations. Section 5 summarise and conclude the research improve quality through DevOps.

II. BACKGROUND AND RELATED WORK

A. SDLC

Software Development Life Cycle (SDLC) is a conceptual model used to describe a process of planning, designing, testing and deploying. Different methodologies exist in the industry to manage SDLC.

The Agile methodology has introduced in response to the drawbacks of the rigid plan driven process models (such as Waterfall, RUP, etc.) and for other resistant methodologies. The Agile methodology is an approach to project management, typically used in software development [1].

The software development market has shown a rapid growth with un-predictions and fast changing. The reasoning behind the rapid changes are client requirements and change requests addressed in positive manner. This has been addressed by agile development and most companies have moved to agile development in order to enable frequent releases with higher customer satisfaction. Though most companies followed agile practices, there were a few companies that used certain agile practices to minimize the bottlenecks on the existing process. Companies such as IBM, Facebook and Microsoft started their own development in Continuous Deployment. As per Claps, Svensson, & Aurum [2] Continuous Deployment is not an easy task since there is significant impact to the system stability. However, Continuous Deployment creates new businesses and also new challenges to software companies [2].

B. DevOps

"DevOps is a mix of patterns intended to improve collaboration between development and operations. DevOps addresses shared goals and incentives as well as shared processes and tools. Because of the natural conflicts among different groups, shared goals and incentives may not always be achievable. However, they should at least be aligned with one another." [3].

Main goal of the DevOps is to identify and eliminate the gaps between the Development team and Operations team. Software developments professionals often tend to come with an outlook where product change is what they are mandated to accomplish. The main goals of DevOps have identified in [4] as following:

- Deliver measurable business value through continuous and high-quality service delivery [4].
- Emphasize simplicity and agility in all areas, including technology, process and human factors [4].
- Breakdown barriers between development and operations by enabling trust and shared ownership, supporting innovation and encouraging collaboration [4].
- Manage dynamic compliance - access/sharing laws are changing [4].

The phoenix project has described the division between following sections when practice DevOps [5].

- Security compliance and IT
- Development and Quality Assurance
- Marketing and IT
- Business goals and IT delivery

Behr and Kim of Phoenix project [5] have noted that DevOps implicitly does not mean that everything will be worked perfectly with frequent deliveries, stable platforms and business IT alignment. It is important to collaborate in place everybody can give their best contribution to resolve problems and support the business [5].

State of DevOps Report 2016 has identified that DevOps has improved organization's performance, revenues and profitability. DevOps teams have increased from 16% in 2014 to 19% in 2015 then 22% in 2016. High performing companies have deployed 200 times more frequently than low performers along with 2,555 times faster lead times. Large companies such as Amazon, Netflix have deployed changes thousands of times per day. These facts imply why companies have moved to practice DevOps [6].

C. CAMS Framework

The CAMS Model has created by DevOps pioneers John Willis and Damon Edward. It stands for Culture, Automation, Measurement, and Sharing. CAMS model has become the model set of values used by many DevOps practitioners. They are the four fundamental and mutually reinforcing values to bring to a DevOps implementation [7].

Culture is the most important entity, which needed to setup prior to implementing DevOps in any environment, where teams work together and share the responsibility for the end users of their application. It does not only encourage the adoption of agile practices in operations work but also allows developers to learn from real world Ops experience and starts a mutual knowledge exchange that break down the walls between teams [8].

Automation is a key enabler for DevOps adoption. The right people, process and tools are needed to create an automation framework for DevOps [7]. Operations and testing teams usually have a good understanding about application performance. Hence, they need to educate developers about importance of maintaining application performance in large-scale environments under heavy load. Providing automated mechanisms to monitor performance in all environments, from CI and test environments to the actual production deployment, allows the shared language of performance to be spoken [8].

Measurements such as business KPIs, system metrics, and application behavior have to be open, transparent, accessible, consumable meaningful, and able to be visualized in an ad hoc manner by all constituents of the DevOps model [7]. With performance aspects being covered in earlier testing stages, performance engineers on testing teams have time to focus on large-scale load tests in production-like environments [8].

Sharing the tools, discoveries, and lessons with others are critical success factors of any organization [7]. Sharing of ideas and problems is the heart of collaboration; it is at the heart of DevOps, too. In DevOps, users must expect to see a high premium placed on openness and transparency.

D. Software Quality

Quality is the most important factor for customer satisfaction and business growth. Quality of the software is measured by software quality characteristics. Software quality characteristics are a set of attributes of a software product by which its quality is described and evaluated. ISO 9126 is an International standard for the evaluation of software quality.

The ISO 9126 software quality model identifies six main quality characteristics, namely Functionality, Reliability, Usability, Efficiency, Maintainability and Portability. Main characteristics are subdivided into twenty seven sub characteristics for internal and external quality [9].

Quality is important for customer satisfaction. Companies need to have an approach align with DevOps to improve the quality. Continuous testing and continuous quality monitoring are completely connected to continuous development, continuous build and continuous deployment

process. It is important to consider Test Driven Development (TDD) and Behavior Driven Development (BDD), which enable common shared understanding of the expected operations and functionality of the application among all team members (users, developers, testers and operations). Quality is not the responsibility of the Quality Assurance (QA) team alone; everyone needs to be responsible for the piece of quality. It is important to consider quality coverage by putting validation checks and create test cases based on the objective rules. It is really important to implement metrics to manage QA efficiency such as iteration performance, release quality which measures defect leakage, number of high severity defects, number of features delivered [10].

Evaluating the Impact of DevOps Practice in Sri Lankan Software Development research paper has identified that Sri Lankan companies has successfully adjust to DevOps environment. Also there is an improvement in responsiveness to business needs, agility to new technologies and quality of the process after starting DevOps [11].

III. Research Model

Based on the key findings of literature review and the main objective of this research, the researchers have designed the conceptual framework in Figure 1. The conceptual framework is divided into two sections, which contains necessary factors to practice DevOps implementation and impact to the software quality.

The hypotheses of the research study have formulated to verify the relationship between the variables. The following hypotheses have derived based on main and secondary objectives of this research.

Hypothesis 1: *Practice DevOps would improve the Quality of the Software*

$H1_0$: There is no relationship between Practice DevOps and Quality of the software

$H1_A$: There is a relationship between Practice DevOps and Quality of the software

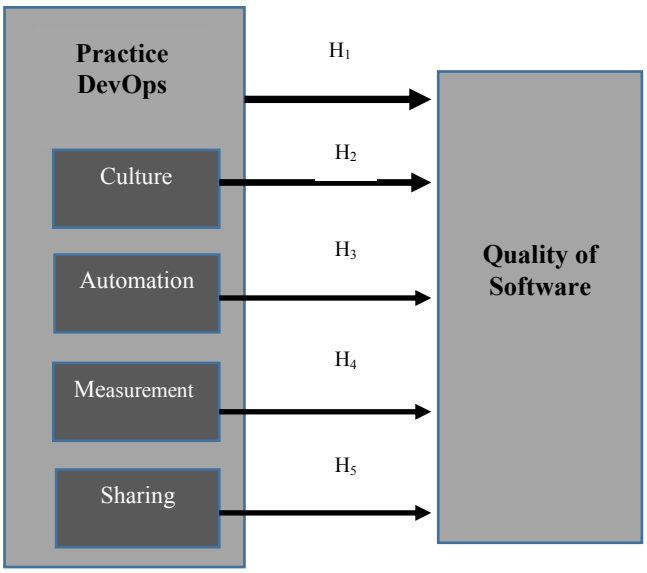

Fig. 1. Research Model and research hypotheses [1]

Hypothesis 2: *Culture of DevOps would improve the Quality of the Software*

$H2_0$: There is no relationship between Culture of DevOps and Quality of the software

$H2_A$: There is a relationship between Culture of DevOps and Quality of the software

Hypothesis 3: *Automation in DevOps would improve the Quality of the Software*

$H3_0$: There is no relationship between Automation in DevOps and Quality of the software

$H3_A$: There is a relationship between Automation in DevOps and Quality of the software

Hypothesis 4: *Sharing in DevOps would improve the Quality of the Software*

$H4_0$: There is no relationship between Sharing in DevOps and Quality of the software

$H4_A$: There is a relationship between Sharing in DevOps and Quality of the software

Hypothesis 5: *Measurement in DevOps would improve the Quality of the Software*

$H5_0$: There is no relationship between Measurement in DevOps and Quality of the software

$H5_A$: There is a relationship between Measurement in DevOps and Quality of the software

Qualitative and quantitative validation of these hypotheses operationalization constructed on 'Improve software quality through practicing DevOps'. The independent variable is 'Practice DevOps', which was measured through Culture, Automation, Measurement and Sharing. Dependent variable of this study was Quality of the software. As shown in Table 1, indicators have been identified based on literature sources to measure the respective variables.

Carefully designed online questionnaire was distributed among software professional, works in companies where practice DevOps. Questionnaire was designed with five point Likert scale questions, open ended question. Also prepared open ended questions to gathered information through face-to-face interviews conducted with DevOps experts.

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	

After prepared the questionnaire researchers have conducted a pilot study. It was conducted for 15 respondent Statistical method Cronbach Alpha has used to select the most suitable question by considering sections Cronbach alpha has above 0.7. This ensures the internal consistency of the variables measured.

There are around 200 Software development organizations registered in Sri Lanka according to Lanka Association of Software Service Companies (SLASSCOM). The companies are categorized in to three different scales: small, medium and large. Respondents were selected based on their DevOps experience, if they have practiced DevOps previously.

The purpose of data analysis is to test the goodness of the data and test the developed hypotheses. Graphical statistical tools such as normality curve and boxplot diagram have been used to test whether data was normal. After confirming that all the variables are normally distributed, Pearson correlation has been used to infer the relationship between independent and dependent variables.

IV. Results and Analysis

A. *Quantitaive Data Analysis and Hypotheses Testing*

An Online questionnaire was distributed among more than 300 software professionals works in companies where practice DevOps and who has prior DevOps experience. 120 responses have been selected for further analysis mainly considering whether they have previous DevOps experience or their companies practice DevOps and based on their professional track.

Figure 2 shows DevOps experience of respondents. 62% Respondents have experience within 2 to 3 years. 24% have experience less than 1 year. 14% respondents have been professional who have practiced DevOps for more than 3 years.

Figure 3 depicts the development methodology used by respondents in their companies. Majority of the respondents (66%) are from companies who practice both DevOps and Agile.

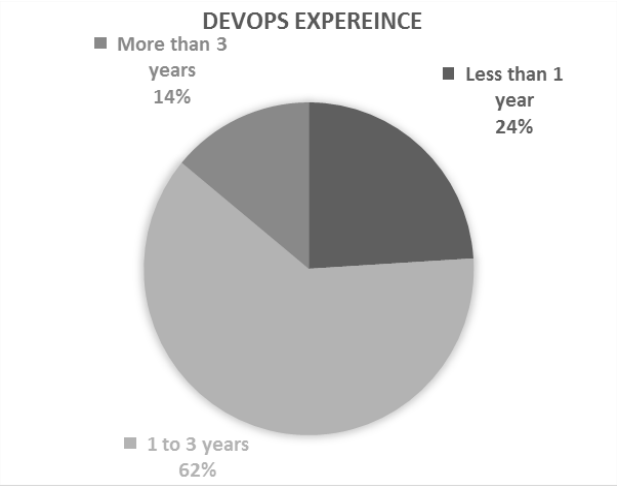

Fig. 2. DevOps Experience

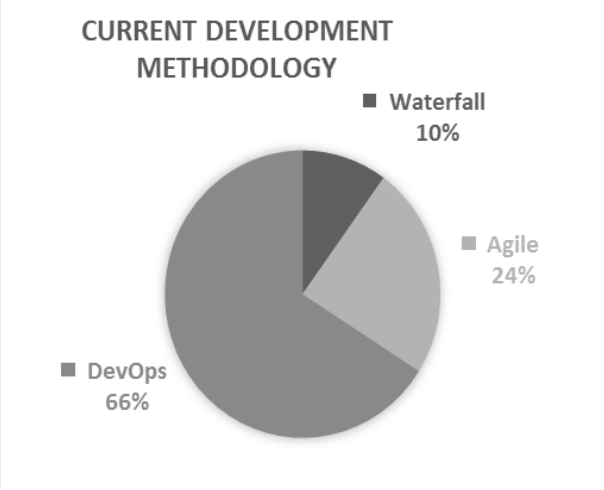

Fig. 3. Current Development Methodology

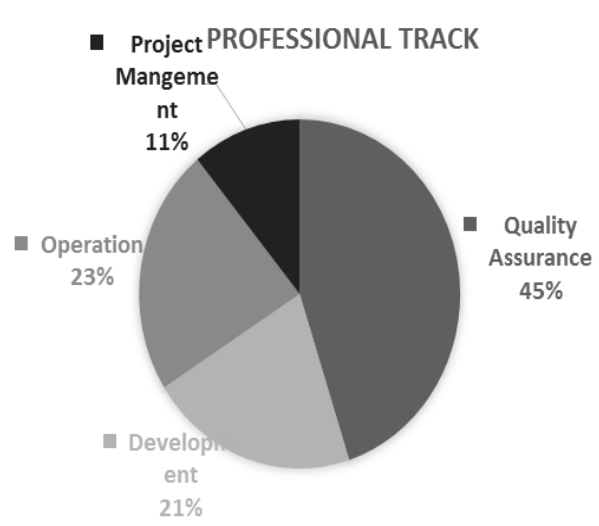

Fig. 4. Professional Track

There have been 24% respondents from companies who practice Agile as the Software development methodology. 10% of respondents have been practiced traditional waterfall methodology but they have prior DevOps knowledge.

Figure 4 illustrates which track respondent included according to their designation. Majority 45% respondents have selected from Quality Assurance track. 21% falls in to Development. 23% respondents are from Operations and remaining 11% is from project management.

The TABLE 2 illustrates the summary of correlation. Relationship between the dependent and independent variables are strongly correlated at 99% confidence level. Based on the table all the five hypotheses have proved with strong positive relationship between dependent and independent variables.

TABLE 2: PEARSON CORRELATION

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

Quality of the software or product is the key success factor of IT business. In this research main objective was to identify the relationship of DevOps practice and how it has impacted on achieving Software quality. Multiple regression analysis has been carried out to come up with model to represent the relationship of DevOps and Quality. Equation is represented with following notation C - Culture, A - Automation, M – Measurement and S – Sharing.

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

According to the above equation, software quality will be increased if A, M, C, and S factors get increased and vice versa. Automation is the critical success factor to improve software quality in DevOps environment. Culture, Sharing and Measure factors also need to be considered to improve the software quality.

B. *Recommendation*

The main objectives of this research is to identify how to increase software quality in DevOps working environment. Qualitative questions and Interview were conducted with DevOps and Quality Assurance experts to identify how to improve the quality after considering factors such as Culture, Automation, Measurement and Sharing.

Quality is important for customer satisfaction. Companies should have an approach align with DevOps to improve the quality. The research findings have indicated that Automation is the most critical success factor to improve quality. Software team first needs to identify Return on Investment (ROI) before start automation. Because some companies blindly do automation, which may not be sustainable in the long run.

Companies need to identify team with automation or capable with technical skills and need to train them to enhance their automation skills. In a situation there is lack of automation resource, management need to recruit employees with required skills as an action item when implementing DevOps.

Test driven development (TDD), behaviour driven development (BDD), and acceptance test driven development (ATDD) are best practices to follow when practice DevOps. TDD is a development practice that starts with writing tests before you write any code. The flow with TDD is that one starts with the desired outcome written as a test. BDD encourages working with the business stakeholder to describe the desired business functionality of the application and expresses the test in a DSL that is pretty close to the natural language. ATDD builds on TDD and BDD, and it is involved in finding scenarios from the end user perspective. Quality engineer has to write automated tests with examples of these use cases, and then run them repeatedly while the code is being developed.

Before start automation it is important set up quality environment to automate the scenarios. There are number of tools to use when practice DevOps such as Jenkins for Continuous Integration, Cucumber for BDD, Junit for TDD, GIT for configuration management, Quality Centre, JIRA, ALM for Test lifecycle and defect management, Selenium, QTP and UFT for automation.

Continuous Integration (CI) is also important when practice DevOps. Automated scripts need to be completed and added to CI environment; as a result, teams can identify whether build is stable and status of the environment.

Culture is another important factor because it changes the way in which teams work together and share the responsibility for the end users of their application. It is important to start mutual exchange with Dev and Ops to break down the wall between teams and both teams will learn from each other mutually. This will depend on working culture of employees based on their society and countries as well.

Company management needs to set up a culture where all employees speak in one language. All teams need to focus on current problems. Finger-pointing between teams need to be avoided and set up them to identify the root cause of production issues or defects. Working together on current issues enables developers, testers, operation teams to become aware of production problems and their solutions; this can be used as risk or mitigation plans future releases.

Performance engineers who are part of quality engineering team have time to focus on large-scale load tests in production-like environments. This helps them to find data-driven, scalability, and third party impacted performance issues. Close collaboration with Ops guarantees that tests can be executed either in the production environment or in a staging environment that reflects production, therefore increasing confidence when releasing a new version.

Earlier in waterfall development methodology testing teams executed performance and scalability tests in their own environments. When working in one environment teams need to share the environment, framework and script with other teams which save time and effort. Sharing knowledge with other teams is an important factor. Quality engineering team can be divided in to functional team, automation team, performance team, accessibility team based on their skill set. Automation team can share knowledge with accessibility team while performance team

can share knowledge with functional team by giving trainings and required knowledge. Therefore sub teams can get help from other teams when they need resources in critical situation.

V. CONCLUSIONS

The main purpose of this paper is to analyse and identify whether software quality gets improved when practice DevOps. Information has gathered through conducting online questionnaires and interviews with DevOps experts in software development industry

After collecting and analysing data it was observed that there is a positive relationship between practice of DevOps and quality of software. Software quality will be increased when practice DevOps. Also there is a strong positive relationship between Culture, Automation, Measurement and Sharing with software quality which mean practice automation, sharing knowledge will increase the software quality.

The output of this research will help companies who practice DevOps and quality engineering teams to take decision to improve testing practices. Research findings have clearly indicated that culture, automation, measurement and sharing have impacted on quality of the products. Therefore practice DevOps will improve the software quality if they consider above facts correctly

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," in *DevOps 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 CommunicationsConference* 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: Reliable Software Releases through Build, Test, and Deployment Automation, Pearson Education, 2010.

[8] , "diazhilterscheid," 28 06 2016. [Online]. Available: hilterscheid.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/downloads/testing/pov_devops-with-quality_ok.pdf. [Accessed 20 01 2017].

[11] P. Perera, I. Perera and M. Bandara, "Evaluating the Impact of DevOps Practice in Sri Lankan Software Development Organizations," *ICT for Emerging Regions,* pp. 281-287, 2016.