



Proposal To Cope Change Resistance Using DevOps

Ahmed Osama Mansour¹; M. Rizwan Jameel Qureshi²

¹Department of Information Technology & King Abdul-Aziz University, Saudi Arabia

²Department of Information Technology & King Abdul-Aziz University, Saudi Arabia

¹ aahmedmansour@stu.kau.edu.sa; ² rmuhammd@kau.edu.sa

DOI: [10.47760/IJCSMC.2020.v09i09.005](https://doi.org/10.47760/IJCSMC.2020.v09i09.005)

Abstract— *This paper is written to support a phenomenon in which development and operations (DevOps) teams work together to deliver software on a continuous pace, eliminating the walls of confusion between stakeholders, and facilitating the business to seize and cope up with existing and constantly emerging opportunities. The main processes are development, quality assurance, pre-operational and operational. There are many tools involved in the implementation of DevOps such as Git, Jenkins, Chef, Vagrant, Docker and Hadoop. Due to the technical expertise in variety of tools, DevOps faces many factors that hinder its adoption such as change in the working norms of traditional teams, team structure and cost. The most significant factor is resistance to change by the traditional teams. In this paper, we will identify the possible change-resistant personnel, the reasons behind change-resistant and a solution is proposed to overcome change-resistant. A survey is used as a research design to validate the proposed solution. The results of survey support the proposed solution that it will help the software companies to cope the change resistance while adopting DevOps.*

Keywords— *DevOps, resistance to change, agile software engineering, automation, people factor*

I. INTRODUCTION

Agile software engineering is one of the hot areas of research from last several years because it promotes to accept changes quickly and efficiently during software development resulting in several potential benefits such as time saving, cost reduction, less resources and high customer satisfaction. Lean and DevOps are the emerging areas of agile software engineering to adapt it for the development of medium and large size projects. Lean concept is taken from manufacturing industry and it is found extremely compatible with agile Scrum methodology. DevOps is a set of practices that provides significant optimization to the software development (SD) process i.e., starting from the creation of ideas and requirements gathering to actually implementing the hardware to run the application. DevOps optimizes the SD process by automating the collaboration between developers and operations teams to deliver software continuously in short releases with high quality. One of the main tasks is to cope resistance to change while implementing DevOps [1]. Software development teams always hesitate to accept new processes and practices that need extra efforts and time due to anonymous risks of failure. There is always a tension between the development and operations teams because of different opinions and goals such as developers are eager to deliver the new code whereas operations team is more concerned with the

quality [2-3]. In this paper, an attempt is made to identify the problems relevant to adopting DevOps with respect to change resistant. The change resistant problem is addressed by proposing a solution to identify the possible personnel obstacles and provide tips to overcome these challenges.

Further paper is arranged as: Section 2 outlines the literature review. Section 3 defines the problem. Section 4 presents the details of the proposed solution. Section 5 validates the proposed solution.

II. RELATED WORK

There are several advantages of the DevOps process such as it improves communications and collaboration among stakeholders [4]. The factor of improved collaboration between development and operation teams helps to ensure continuous and smooth delivery to clients. DevOps also helps SD companies to deal easily with emerging market's requirements and it cuts short customer's feedbacks cycle. There is not enough effort made in majority of the software companies to implement DevOps adoption strategies. Consequently, DevOps is not being adequately communicated and the desired benefits are not achieved. A study has been conducted to investigate the reasons behind the hindering of DevOps adoption. To identify the DevOps hindering factors, several interviews have been conducted to actively involved DevOps practitioners. Lack of training, risk of losing authority, resistance to change, lack of long term direction from senior management and silo mentality are the main factors hindering the software development companies to implement DevOps. It is concluded from the study that social values has a dominating factor in most of the success stories that are reported about DevOps adoptions [3]. A major criticism, to the study of Kamuto and Langerman [4], is that it does not provide practical evidences to validate. There is a need to generalize the results because the case study was conducted in South Africa.

Gupta et al. [5] present their experience of working in a SD project in which distributed teams are involved based in USA, India, and Germany. It is mentioned that the team has successfully established continuous delivery cycle of short releases using DevOps and agile Scrum methodology. There are three main challenges reported to integrate agile and DevOps i.e., how to convert: 1) fixed scope based releases into time boxed release strategies; 2) traditional milestone based execution into value stream based execution and 3) manual processes into automated processes from development to deployment. To address the challenges of integrating DevOps and agile Scrum, there are several recommendations reported in the study of Gupta et al. [5] such as organize seating of developers and operational staff close to improve their collaboration, create a Q-interface, that is a type of mapping documents for common quality documents, in case of distributed teams and automate the process of test, deployment and operation.

Software logging plays a critical role to ensure the quality of the deployed software using DevOps [3]. Chen [3] proposes the automated approaches to improve software logging practices. DevOps logging challenges are summarized as follows. According to Chen [3],

- “Challenges 1 and 2 are reported on the development side
 - No existing guidelines on producing high-quality logging code.
 - Difficulty in maintaining and evolving logging code.
- Challenges 3 and 4 are described on the operation side
 - Limited mechanism for quality feedback.
 - Heterogeneous and complex telemetry data.”

To tackle these challenges, four software repositories are recommended i.e., historical, communication, runtime and bug [3]. Historical repository refers to the source code version control systems like GitHub and SVN. The communication repository will be used for the online communication. The runtime repository is used to store telemetry data generated in various scenarios. The bug repository is used to track the issues related to bugs and project such as JIRA and Bugzilla. For each challenge, a solution is proposed as follows.

- Filter and analysis the problems of logging.
- Estimate the impact of various logging techniques.
- Measure code coverage via execution logs.
- Relating telemetry data to fix the problems.

It is highly recommended to utilize security practices while using DevOps [4]. A survey is conducted in nine software development organizations, opted DevOps, to conclude the recommendation. It is also reported that automated monitoring activity is beneficial to the security of software developed in a DevOps environment. There are only nine organizations involved while conducting the survey therefore it is hard to generalize the findings [4]. There is also no discussion about the automated activities of DevOps on the quality of resultant software.

In the past few years, an increase in the adaptation of DevOps is observed due to the fact that it improves collaboration between development and operations teams resulting in speedy delivery of high quality software [5]. One of the main technologies, essential to implement DevOps, is the use of infrastructure as code (IaC) scripts such as Puppet [5]. SD organizations use IaC scripts to automate its operations and configuration of infrastructure. IaC scripts can contain defects resulting in dire consequences such as consistency of code in the GitHub. Defect prediction could help software teams to prioritize inspection efforts. Rahman et al. [5] proposed defect prediction model-related metrics to help software practitioners in prioritizing inspection efforts for IaC infrastructure scripts. A process is discussed to identify metrics that characterize defect in IaC scripts using Constructivist Grounded Theory. Eighteen metrics are identified and each of these metrics corresponds to a specific characteristic of defects in IaC scripts. The metrics are categorized into three types i.e., Churn-related metric (CH), IaC-related code metric (CO) and generic metric (GE). There is a further need of a study to validate the eighteen metrics because only three datasets are used.

DevOps provides automation to increase the speed of software delivery but there is a great challenge to maintain the momentum of speedy delivery while performing security activities in parallel [6]. A key security enabler is threat modeling acquiring highly skilled individuals. Valani [6] proposes a lightweight threat modeling approach relying on a correlation matrix. The correlation matrix mapped security requirements over application archetypes. Thereupon, the result has been translated and it is claimed that the proposed lightweight threat modeling approach removed many DevOps security bottlenecks.

The problem in most of the agile SD companies is that they focus more on the development side rather than the operation side [7]. DevOps is introduced to overcome this problem and it breaks the barrier between developers and operations teams. However, both sides have to pay attention to bridge the communication gap especially in case of distributed teams. An exploratory observational study is conducted in a multinational company opted DevOps. The study is based on the data that is gathered from distributed teams. The development team is located in Australia and the operational team is based in Brazil. Seven main challenges are reported for the successful implementation of DevOps. These challenges are geographical distance, socio-cultural distance, temporal distance, communication frequency, direction of communication, modality, and content. The results cannot be considered reliable because these are mainly concluded from the interviews [7].

DevOps is gaining popularity due to its ability to deliver software continuously and ease of adaptation in case of changing requirements. Hussain et al. [8] investigate online job advertisements and interviews to identify key knowledge, capabilities, and skills of DevOps teams. The investigation findings show that “DevOps being adopted as a philosophy whereby responsibilities of individual members are shared across increasingly joint Development and Operations teams”. The investigation is carried out on DevOps jobs in the SD companies of New Zealand with collocated teams therefore it is difficult to generalize the results in other settings. DevOps methodology implementation does require considering security quality standards but the problem appears when SD company is trying to maintain delivery velocity and adhere to security standards [9]. The investigation identified that the SD organizations have to adjust their infrastructure according to DevOps methodology before taking its advantage of speedy delivery.

In the past few years, many enterprises have adopted continuous integration practice of agile practice in their software delivery lifecycle [10]. It is emerged from last few years that continuous integration practice is not enough to consistently maintain the momentum of delivery cycle. This is a dilemma for the SD companies because the development teams are delivering at a much faster pace than the operational teams are deploying resulting in a gap between both teams. Virmani [10] claims that the DevOps methodology is one of the solutions to bridge the gap between the development and operational teams.

The main limitations of related work are shown in Table 1.

III. PROBLEM STATEMENT

DevOps is one of the emerging technologies opted in the software industry from the last few years due to several potential benefits such as time saving, within budget and high quality products with short release cycles. It is not easy to ascertain the benefits of DevOps implementation as it demands to change the working environment. The working environment needs highly skilled, motivational and synchronized people. The people are reluctant and resistant to change their behaviors and adapt the requirements of DevOps implementation in terms of new roles and responsibilities [11]. The research question is narrated as follows.

- How to overcome change resistance to implement DevOps?

IV. THE PROPOSED SOLUTION

The proposed solution defines two goals to address the change resistant to achieve successful adaptation of DevOps in a SD company to achieve win-win.

Goal A- Identify the challenges of change-resistant people.

Goal B- Address the challenges of change-resistant people.

A. Identify the challenges of change-resistant people

Project manager (PM) is the one of the team members who is resistant and affected the most while implementing the DevOps concept. The reasons behind the resistance of project manager are due to his new responsibilities and required abilities such as clarify the DevOps necessity to all stakeholders, familiar with DevOps techniques and tools and support collaboration between the quality assurance, development and operations teams. DevOps implementation ensures frequent releases in a short span of time whereas the quality assurance (QA) teams require ample time to apply possible test cases in the traditional system development life cycle. The QA teams are also reluctant to implement DevOps due to learning new skills and tools to perform automated testing. The development teams are also resistance to opt DevOps because they need to work along with the operations team and caring about infrastructure issues those are cumbersome to make software operational. The operational teams are also reluctant to be a part of DevOps. This is because they have to coordinate with developers and quality assurance engineers and regularly status reporting to project manager. Also, the new roles of operational teams demand to get familiar with version control and automation tools.

B. Address the challenges of change-resistant people

To address the challenges of change resistant PM, it is recommended in this research to train him adequately in all areas of DevOps environment such as process, tools, continuous integration, continuous delivery (CI/CD), containerization, cloud technologies and infrastructure as code (IaC). PM is also responsible to create a collaborative environment to control and manage developers, operations and quality assurance teams. It will enable a PM to accomplish projects successfully following DevOps. PM has to foster developers' soft skills to create a mutual harmony between developers and operations teams in order to make sure that the operations team can work with the tools used by the developer team e.g. automation server (Jenkins) and containerization tools (Docker). To decrease the efforts consume in performing manual testing, a PM should collaborate with the QA manager to clarify his concerns with respect to DevOps. QA team needs adequate training in API automated testing tools. PM has to use his soft skills in order to increase the operations team's desire to work closely with developers and quality assurance teams.

Fig. 1 shows the proposed solution to address the change resistant to implement DevOps.

TABLE I
MAIN LIMITATIONS OF RELATED WORK

Paper Title	Limitation
Factors inhibiting the adoption of DevOps in large organisations: South African context [1].	The researchers didn't provide practical evidence on their proposed framework validity. So, there is a need for evidence of the presented conceptual framework validity. Also, the use case was held in the South Africa domain, so to some extent, we can't generalize the findings on the rest domains.
Challenges in Adopting Continuous Delivery and DevOps in a Globally Distributed Product Team: A Case Study of a Healthcare Organization [2].	The organization was already implementing agile methodology in their development before converting to DevOps, the researchers didn't mention the case of coming from different development backgrounds which may differ the whole process.
Improving the Software Logging Practices in DevOps [3].	The thesis provides an empirical study, so it may require practical evidence to confirm its efficiency.
Software Security in DevOps: Synthesizing Practitioners' Perceptions and Practices [4].	The number of surveyed organizations is small; we could not claim findings are generalizable, and there is no examination of the relationship between the use of automation activities, and the quality of deployed software.
Poster: Defect Prediction Metrics for Infrastructure as Code Scripts in DevOps [5].	The chosen eighteen metrics were not comprehensive and only three datasets were used to evaluate the methodologies which are not enough for generalization.
Rethinking Secure DevOps Threat Modeling: The Need for a Dual Velocity Approach [6].	It's claimed that the proposed lightweight threat modeling approach removed many secure DevOps bottlenecks. Although a practical evidence is needed.
Communication Challenges and Strategies in Distributed DevOps [7].	A small sample of DevOps employees were used from only one company. Therefore, the result may vary from different teams, putting into consideration their different contexts and cultural backgrounds. Also, results rely only on observations confirmed by interviews, which may affect reliability.
Emerging Trends for Global DevOps: A New Zealand Perspective [8].	The researcher assumed the content of online job postings is valid as a representation of employers' specifications.

	Also, investigation held on DevOps jobs in the New Zealand domain; it is difficult to generalize the results.
Compliance at Velocity within a DevOps Environment [9].	Data collection is not sufficient, observing an environment that fully implemented DevOps and technical compliance testing could improve data collection.
Understanding DevOps & bridging the gap from continuous integration to continuous delivery [10].	A practical evidence is needed to validate the proposal.

V. VALIDATION

A survey is conducted to evaluate the proposed solution. A questionnaire is designed that is consisting of nineteen close ended questions those are further divided into two goals. A sample size of thirty participants is collected against the two goals to statistically analyze and interpret. Goal 1 identifies the possible change-resistant personnel. Goal 2 overcomes the change-resistant for each personnel. Likert scale, one to five, is used to evaluate the questions from very low to very high. The results are shown in the form of frequency tables and bar charts.

Thirteen questions are designed to evaluate the goal 1. Table 2 shows the cumulative descriptive analysis of goal 1. Six questions are asked to evaluate the goal 2. Cumulative results of six questions are shown in Table 3.

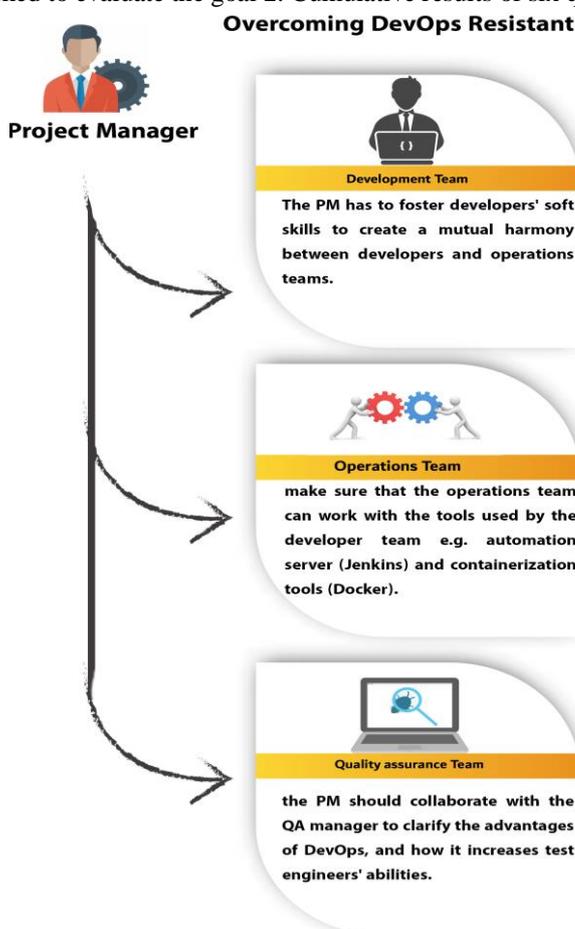


Fig. 1 The proposed solution to manage change resistant using DevOps

A. Cumulative Statistical Analysis of Goal 1

Table 2 shows that 27.5% of the participants are strongly agreed, 38.3% are agreed, 23% are Neutral, 10% are disagreed and 3% are strongly disagreed. Fig. 2 displays the results graphically.

TABLE II

CUMULATIVE ANALYSIS OF GOAL 1

Q. No.	Very Low	Low	Nominal	High	Very High
1	1	18	1	3	7
2	0	0	2	17	11
3	11	8	1	4	6

4	0	1	0	19	10
5	0	1	6	15	8
6	0	0	2	20	8
7	0	1	2	18	9
8	1	11	4	5	9
9	2	16	2	2	8
10	0	1	0	18	11
11	2	12	2	4	10
12	11	2	3	6	8
13	1	3	1	15	10
Total	28	74	28	146	105
Avg. %	7.3	19.4	7.3	38.3	27.5

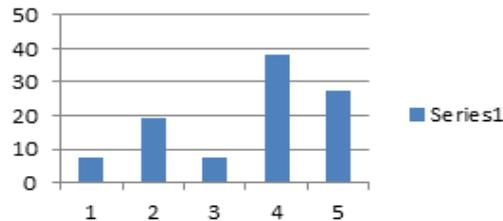


Fig. 2 The cumulative analysis of goal 1

B. Cumulative Statistical Analysis of Goal 2

Table 3 shows that 26.6% of the respondents are strongly agreed, 49.4% are agreed, 22.7% are neutral, 0.5% are disagreed and 0.5% are strongly disagreed. The graph in fig. 3 appears to be supporting the suitability of goal 2.

TABLE III
CUMULATIVE ANALYSIS OF GOAL 2

Q. No.	Very Low	Low	Nominal	High	Very High
14	0	0	7	20	3
15	1	0	6	16	7
16	0	1	8	16	5
17	0	0	6	15	9
18	0	0	7	13	10
19	0	0	7	9	14
Total	1	1	41	89	48
Avg. %	0.5	0.5	22.7	49.4	26.6

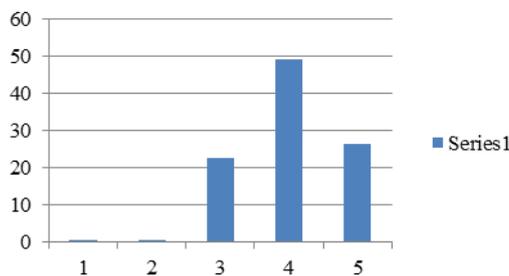


Fig. 3 The cumulative analysis of goal 2

C. Final Cumulative Statistical Analysis of Two Goals

The final cumulative evaluation of two goals is shown in Table 4. Table 4 shows that 27.1% of the respondents are strongly agreed, 43.9% are agreed, 15% are neutral, 9.9% are disagreed and 3.9% are strongly disagreed. The graph in fig. 4 appears to be supporting the suitability of two goals.

TABLE IV
CUMULATIVE ANALYSIS OF TWO GOALS

Goal No.	Very Low	Low	Nominal	High	Very High
Goal 1	7.3	19.4	7.3	38.3	27.5
Goal 2	0.5	0.5	22.7	49.4	26.6
Total	7.8	19.9	30	87.7	54.1
Avg. %	3.9	9.9	15.0	43.9	27.1

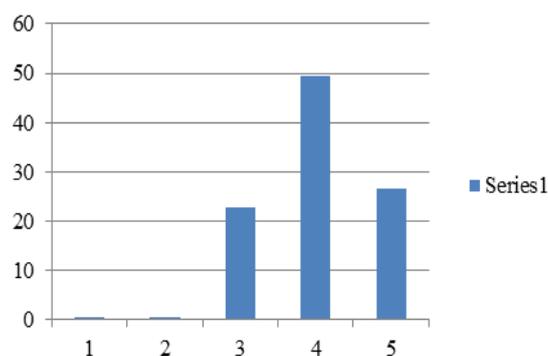


Fig. 4 The cumulative analysis of two goals

VI. CONCLUSION

DevOps is a methodology where the development of a product or service is integrated with its operation. In DevOps, the development and operations teams are broken down into smaller groups consisting of members from both teams. This integration allows teams to use methods such as continuous integration and continuous delivery (CI/CD) which allows them to deliver new features to market faster and fix bugs more efficiently. When it is time to adopt the DevOps methodology, each personnel, involved in the software development life cycle, has his own reasons to resist against the DevOps methodology such as new roles and duties, uncomfortable to work closely with other teams and apply DevOps technology. In order to overcome the resistance, the company's chief information officer (CIO) has to establish open communication between each staff member involved in the SDLC and clarify their new duties and take notes of their concerns. It is also necessary to train the team in order to explain the advantages and changes that DevOps will bring into the SDLC.

REFERENCES

- [1] M. B. Kamuto, and J. J. Langerman, "Factors inhibiting the adoption of DevOps in large organisations: South African context," in *Proc. RTEICT'17*, 2017, p. 48-51.
- [2] R. K. Gupta, M. Venkatachalapathy and F. K. Jeberla, "Challenges in Adopting Continuous Delivery and DevOps in a Globally Distributed Product Team: A Case Study of a Healthcare Organization," in *Proc. ICGSE'19*, 2019, p. 30-34.
- [3] B. Chen, "Improving the Software Logging Practices in DevOps," in *Proc. ICSE '19*, 2019, p. 194-197.
- [4] A. A. Rahman, L. A. Williams, "Software security in DevOps: synthesizing practitioners' perceptions and practices," in *Proc. CSED'16*, 2016, p. 70-76.
- [5] A. Rahman, J. Stallings and L. Williams, "Poster: Defect Prediction Metrics for Infrastructure as Code Scripts in DevOps," in *Proc. ICSE'18*, 2018, p. 414-415.
- [6] A. Valani, "Rethinking Secure DevOps Threat Modeling: The Need for a Dual Velocity Approach," in *Proc. SecDev'18*, 2018, p. 136-136.
- [7] E. Diel, S. Marczak and D. S. Cruzes, "Communication Challenges and Strategies in Distributed DevOps," in *Proc. ICGSE'16*, 2016, p. 24-28.
- [8] W. Hussain, T. Clear and S. MacDonell, "Emerging Trends for Global DevOps: A New Zealand Perspective," in *Proc. ICGSE'17*, 2017, p. 21-30.
- [9] M. Z. Abrahams and J. J. Langerman, "Compliance at Velocity within a DevOps Environment," in *Proc. ICDIM'18*, 2018, p. 94-101.
- [10] M. Virmani, "Understanding DevOps & bridging the gap from continuous integration to continuous delivery," in *Proc. INTECH'15*, 2015, p. 78-82.
- [11] C. Craven (2019) What is DevOps? Definition. [Online]. Available: <https://www.sdxcentral.com/industry/devops/definitions/what-is-devops-definition/>