IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 16, 2020, accepted May 26, 2020, date of publication June 1, 2020, date of current version June 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2998819

Prioritization Based Taxonomy of DevOps Security Challenges Using PROMETHEE

SAIMA RAFI[1], WU YU[2], MUHAMMAD AZEEM AKBAR[3], AHMED ALSANAD[4], AND ABDU GUMAEI[4]

[1]School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[2]School of Cyber Security and Information Law, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[3]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
[4]STC's Artificial Intelligence Chair, Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh 11451, Saudi Arabia

Corresponding authors: Saima Rafi (saeem112@yahoo.com), Wu Yu (wuyu@cqupt.edu.cn), and Ahmed Alsanad (aasanad@ksu.edu.sa)

This work was supported by the National Social Science Foundation of China under Grant 17XFX013.

ABSTRACT DevOps is a combination of collaborative and multidisciplinary efforts of an organization to control continuous delivery and updates of new software while guaranteeing their reliability and correctness. In the software industry, the implementation of DevOps (development and operations units) faces many challenges that are specifically associated with the security. The objective of this study is to identify and develop a prioritization based taxonomy of DevOps security challenges. The total of eighteen DevOps security challenges were extracted using systematic literature review approach and were further evaluated with experts using questionnaire survey study. Finally, the multi criteria decision making PROMETHEE-II approach was used to prioritize and develop the taxonomy of identified factors and their categories. The implications of PROMETHEE-II approach are novel in this research domain as it has been used successfully in various other domains e.g. medical, banking, internet techniques and management etc. The contribution of this study is not limited to develop the taxonomy based structure of DevOps security challenges, but also the proper prioritization of these challenges by introducing PROMETHEE-II approach in the research field of DevOps. The study results will assist the practitioners to remove the uncertainty and vagueness in the opinion of DevOps experts to secure DevOps implementation for better and continuous software development process.

INDEX TERMS DevOps security, challenges, empirical investigations.

I. INTRODUCTION

DevOps is a new paradigm that focus on the collaborative and multidisciplinary nature of organization to control automated delivery and updates of software while guaranteeing their effectiveness. In software industry, the DevOps is a trending technology that focus on collaboration between and within teams involved in software development. It refers to improving the performance of software enterprises (continuous deployment) by coordinating the development and operation teams in one process [1]. Software enterprises need to adapt their protocols and practices to the various modifications brought about by new technology concepts such as "DevOps" [2]. According to Puppet Lab 2015 report, the enterprise with DevOps environment experienced 30 times more deployment rate then the enterprises who have not adopted DevOps in their software development cycle. Therefore, to compete with software industry pillar, the enterprises are moving towards DevOps [3]. CA Technology [4] put forward their research regarding DevOps adoption, coming out with outcomes that 88% of 1425 surveyed software enterprises will move their trend towards DevOps in next five years.

Despite the popularity of DevOps, security is among the major concerns that hinder the adoption of DevOps in software enterprises [5]. This triggered the mapping of security practices with the DevOps process by promoting the security team to collaborate with development and operation teams. Rahman and Williams [6] also emphasizes on the importance of security, by breaking silos of security, sharing that knowledge with various teams of software development process in order to build the relationship between them.

The associate editor coordinating the review of this manuscript and approving it for publication was Porfirio Tramontana.

Security must be treated forefront in an enterprise in-order to adopt DevOps successfully. The deployment rate of software enterprises increasing day by day e.g. Facebook deployment rate is 500 times per day [7]. Therefore, to maintain such rapid rate if DevOps team will work without coordination with the security team, they might not get the desired output in terms of secure software. The new production unit has more chances of vulnerabilities, which can only be controlled by rapid action required by collaboration of DevOps with security practices [6]. By working as, a one-unit team DevOps process will help software enterprise in achieving better quality of software [8]. These concerns motivated us to identify security challenges in software enterprises with respect to DevOps, aiming to assist the software practitioners to move in the direction of secure DevOps adoption.

Due to the increasing popularity of security within DevOps, in this paper, empirical study has been conducted to address the concerns of software practitioners in terms of secure DevOps environment. Therefore, the significance of DevOps motivated us to explore and analyze security challenges in DevOps faced by practitioners by conducting a detailed empirical study. To achieve this study objective, firstly, we will perform a systematic literature review to identify the security challenges from literature and check the validity of identified challenges in real industry considering questionnaire survey approach. Secondly, we will apply Preference Ranking Organization Method for Enrichment Evaluation (PROMETHEE) technique to prioritize the identified challenges considering their significance to DevOps.

There are many approaches of multi criteria decision-making methods used for prioritization in different fields of engineering. For example, (ELECTRE, AHP, WSM, TOPSIS, and PROMETHEE etc.). However, ElECTRE and PROMETHEE are outranking models, which compares the alternative pairwise for each criterion, finding the strength of preferring one to the other [84]. The PROMETHEE II method has been used by scholars in other fields of information technology, especially computer science, management and internet techniques where multi criteria decision making is required [9], [10], [13]. This method has an appropriate structural system and the results are more consistent, easy to understand and require less information as compared with AHP, ELECTRE [10], [83]. For example, Veza *et al.* [11] prioritized the industrial enterprise based on the enterprise's competences. They designed a special set of competence against each enterprise for evaluation purpose and rank them by applying the PROMETHEE II method. Theodorou *et al.* [12] applied PROMETHEE II approach to analyze the decision of Cyprus energy resources. Siahaan and Mesran [9] applied PROMETHEE II ranking technique to determine the best student at college using various attributes like skills, performance, grading etc. Liu and Guan used PROMETHEE II method to evaluate the quality of railway passenger service. The linguistic variables were transformed to fuzzy triangular scale and based on evaluation process the quality service was prioritized [10]. Zhao *et al.* [13] modified PROMETHEE II to provide timely evaluation of incident management plans e.g. earthquake, flood etc.

We believe that in-depth study of security challenges in DevOps will help software industry practitioners to revise their practices and develop a new progression cycle for success and development of secure DevOps practices. To meet this study objective, the following questions were designed.

[RQ1]: What are the security challenges of DevOps implementation reported in literature?

[RQ2]: Are the identified DevOps security challenges related to industry practitioners?

[RQ3]: How can we prioritize the identified challenges?

[RQ4]: How can we develop a prioritization-based taxonomy of the investigated challenges?

II. BACKGROUND OF DEVOPS SECURITY

Recent studies have focused on the importance of DevOps and recognized that, to streamline the software development cycle in terms of better performance and scalability, developers and operation teams must tune-up. This trend of coordination (development and operation teams) at real time enables the software production system to monitor and react whenever anomalies are detected [15]. Lack of correlation activities between development and operation teams cause challenging problems which include: (1) poor information and communication flow, (2) security related concerns, (3) immature systems, (4) unsatisfactory test environment etc. [14], [16].

To meet the on-demand infrastructures and continuous delivery product, DevOps software organizations are searching for active development approaches. There are many challenges of DevOps identified in literature such as "lack of DevOps knowledge" [69], "conceptual gap between development and operation team" [70], "lack of efficient tools" [20], "continuous testing" [3] etc. However, prior work is done in domain of DevOps security. Combining the expertise of development, operations and security within DevOps environment can resolve several security issues. Researchers from academia and industry agree to integrate security practices in the DevOps environment. For example, Cash *et al.* [18] imply to incorporate the security practices with DevOps as SecDevOps. Rahman *et al.* [17] stated that "DevSecOps, SecDevOps, SecOps, and RuggedOps are aliases of Security in DevOps. These terms refer to the integration of security principles in DevOps by promoting the collaboration between the security teams, the development teams and operations teams." They also believe that automated testing and monitoring contributes positively while dealing with software security. Vries [19] emphasized on how the traditional security practices, with a focus on manual processing tools and documentation, are unfit in an environment of continuous deployment. The security practices must be replaced by more upgrade approaches in order to meet the requirements of continuous deployment process.

Prior studies have discussed security aspects with respect to DevOps and agile practices in software organizations.

Smeds *et al.* [20] highlighted the key concerns amongst organizations for adopting DevOps, but did not consider the security challenges while working in DevOps organization. Rahman and Williams [3] also discussed security in DevOps by adding additional security activities such as, security requirement analysis and security configurations with DevOps to remove vulnerabilities. Mohan and Othmane [5] investigated the main security aspects related to DevOps i.e. 'definition, best practices, configuration methods, tools of SecDevOps, team coordination and data secrecy activities' and give a suggestion of merging security with DevOps, as security is one of the key challenges which limits the adoption of DevOps. Therefore, while dealing with DevOps progression, the security must be considered as an essential part of development. In our study, we have listed the security challenges in DevOps that can contribute by helping software practitioners to upgrade their security practices related to DevOps. Furthermore, by introducing the PROMETHEE II [21] approach in the research field of DevOps we prioritized the challenges which will assist organizations to remove vagueness in the opinion of experts to secure DevOps. Our findings will also provide a taxonomy based on CAMS (DevOps principles) [22], [23], [25], which could help experts to improve their management strategies to secure DevOps continuous activities.

III. RESEARCH DESIGN

To achieve the study objectives following three research methodologies were applied (Figure 1):

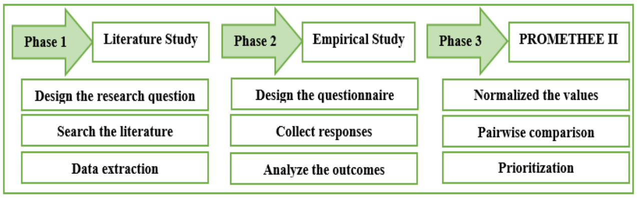


FIGURE 1. Proposed methodology flow.

PHASE 1: Identify the DevOps security challenges in literature using systematic literature review.

PHASE 2: The questionnaire survey to empirically validate the identified challenges of DevOps security from industrial perspective.

PHASE 3: PROMETHEE II approach to prioritize the identified challenges concerning their importance for DevOps security.

A. SYSTEMATIC LITERATURE REVIEW (SLR)

We have adopted systematic literature review approach to explore the challenges of DevOps security in literature. This step by step approach of literature collection, examination and extraction of data gives more valid outcomes as compared with other informal methods of literature review. The guidelines provided by Kitchenham and Charters [64] was adopted to explore literature which include i) planning the review, ii) conducting the review and iii) reporting the review. The phases are discussed briefly in subsection below:

1) PLANNING THE REVIEW

This phase consists of following steps:

a: RESEARCH QUESTION

The purpose of conducting SLR is to explore literature to collect factors that have a negative impact on DevOps security. For this we have designed a following research question.

RQ1: What are the security challenges of DevOps implementations reported in literature?

b: DATA COLLECTION SOURC

Data collection is an important step while selection of authentic data sources. Therefore, we have considered the recommendations of Chen *et al.* [65]. The selected data repository includes:

1. http://ieeexplore.ieee.org
2. http://dl.acm.org
3. www.wiley.com
4. http://link.springer.com
5. http://scholar.google.com
6. https://digital-library.theiet.org

c: SEARCH STRING

The search string plays an important role in the collection of data from selected studies. We have also developed the search strings using key terms and their alternatives collected from other studies by using the Zhang *et al.* [66] guidelines. The key words and their alternatives are given below:

("barriers" OR "obstacles" OR "hurdles" OR "difficulties" OR "impediments" OR "hindrance" OR "concerns" OR "challenge") AND ("SecDevOps" OR "DevSecOps" OR "SecOps" OR "security" OR "privacy") AND ("DevOps" OR "Development and Operation", OR "Continuous development and operation").

d: INCLUSION AND EXCLUSION CRITERI

Protocols were designed to perform inclusion and exclusion criteria on literature collected in response of search strings. The same approach has been adopted in other software engineering researches as Niazi *et al.* [42] and Akbar *et al.* [44]. The considered protocols are presented below:

Inclusion Criteria:

- The paper should be published in the well reputed journal, white paper, book or conference.
- The article must consist of factors that have a negative influence to secure DevOps.
- The paper should have a clear concept about DevOps implementation.
- The selected article must be in English language.

Exclusion Criteria:

- If two studies are from similar project only the most complete one will be considered.

- The paper that does not provide detailed information about DevOps progression.
- The studies not related to DevOps security will not be considered.
- The literature studies are not considered.

e: STUDY QUALITY ASSESSMENT (QA)

In this step we perform QA to check the effectiveness of study with respect to the research objective. We have followed the Kitchenham and Charters [64] guidelines for QA. The five scale Likert scale was used for this assessment. The detail analysis of QA with developed research questions and scale is given in Appendix C (Table 15).

2) CONDUCTING THE REVIEW

a: FINAL STUDY SELECTION

For further refinement of selected literature, we have adopted tollgate approach developed by Afzal *et al.* [67]. In the initial stage after performing inclusion and exclusion criteria we have collected 110 studies. All the steps of tollgate approach were performed carefully and total of 40 studies were collected for data extraction (Figure 2). We have performed the QA process, to check the effectiveness of studies briefly discussed in Appendix C (Table 15).

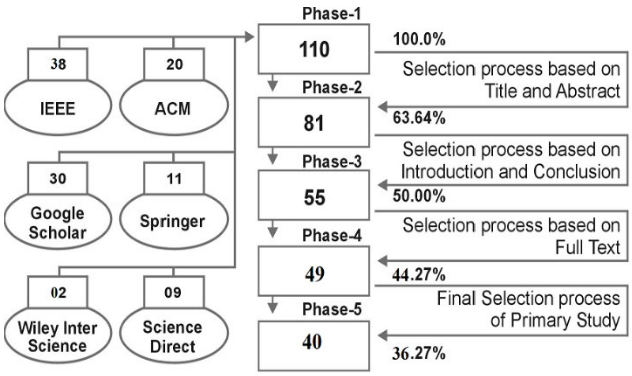

38
IEEE
20
ACM
30
Google Scholar
11
Springer
02
Wiley Inter Science
09
Science Direct
Phase-1
110
100.0%
Selection process based on Title and Abstract
Phase-2
81
63.64%
Selection process based on Introduction and Conclusion
Phase-3
55
50.00%
Selection process based on Full Text
Phase-4
49
44.27%
Final Selection process of Primary Study
Phase-5
40
36.27%

FIGURE 2. Tollgate approach steps.

b: DATA EXTRACTION AND SYNTHESIS

The data extraction is performed clearly by involving first three authors of this study. In initial step the themes, concepts and challenges of DevOps security were extracted from the selected studies. After synthesizing the data, we have total eighteen security challenges of DevOps. There may be biases between the study findings, to remove this concern we have conducted inter-rater reliability test [68]. We invited three external experts to participate in the data validation process. They performed all steps of data extraction by randomly selecting ten studies. After comparing the outcomes from external experts and study authors, we have calculated the Kendalls coefficient of concordance (W) [68]. The value ‘W = 0’ represents complete disagreement and ‘W = 1’ represents complete agreement. The results of W = 0.84, (p = 0.003) show the significance agreement between the external experts and study authors.

The code used to calculate (W) is given in link https://rdrr.io/cran/DescTools/man/KendallW.html.

3) REPORTING THE REVIE

a: QUALITY OF SELECTED STUDIES

The quality assessment is performed to measure the significance of selected literature to address the research question of this study. The 65 % of studies score more than 60% (Appendix C) which shows that the selected studies are significant enough to answer the research question of this study. We choose 50% as a threshold value while performing QA.

b: PUBLICATION YEARS OF SELECTED STUDIES

During the data extraction phase we have also extracted the publication year with the aim to check the frequency of publications literature about DevOps. The frequency analysis shows that the selected studies are from year 2011 to 2020 showing the increasing trend of research in the field of DevOps. This shows that the DevOps is an active topic in the research field of software engineering.

B. EMPIRICAL STUDY

To validate the findings of SLR and to identify more challenges of security in DevOps we have conducted a questionnaire survey approach Figure 3. This approach will assist to collect responses from industrial practitioners working to secure DevOps activities. The questionnaire approach provides the best opportunity to collect practitioner’s opinion from large population [31]–[35].

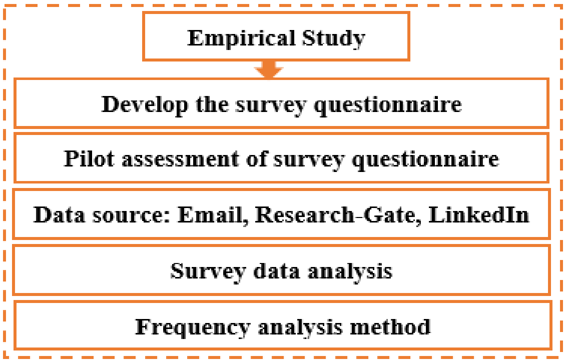

Empirical Study
Develop the survey questionnaire
Pilot assessment of survey questionnaire
Data source: Email, Research-Gate, LinkedIn
Survey data analysis
Frequency analysis method

FIGURE 3. Empirical study flow.

1) QUESTIONNAIRE DEVELOPMENT

The design of the questionnaire was developed by using the platform of Google form (i.e. docs.google.com/forms). The questionnaire consists of three sections, that includes i) bibliographic information of survey participants, ii) the second section is about closed ended questions that consists of DevOps security challenges identified from SLR study, iii) the third section consists of open-ended questions to ask participants about additional security challenges of DevOps.

2) PILOT ASSESSMENT OF QUESTIONNAIRE SURVE

The pilot assessment of questionnaire was performed with aim to check the understandability and suitability of designed questionnaire [36].

The designed questionnaire was sent to three external experts of academia and industry for evaluation. The requested experts include two industrial experts from (Virtual force Pakistan and I-Tech Malaysia) and one from academia (King Fahad University of Petroleum and Minerals, Saudi Arabia). Some modifications were suggested by the participants related to the structure of the questionnaire and to add question regarding DevOps experience in an organization. They further suggested to use tabular format for the questions response. We carefully reconstruct our questionnaire survey, according to the recommendations suggested by participants, and final format of questionnaire is given in Appendix A.

3) DATA SOURCE

The goal of this study is to identify the security challenges in DevOps. The data source plays an important role to get an opinion from targeted population. Though, to validate these identified challenges from the target population we have used Research- Gate, Emails and LinkedIn profiles. To spread the questionnaire survey, we have used the snowball technique [37], [38], [40] which is cost effective and easy way to target large scale population. The duration of data collection is from November 2019 to January 2020. We have collected total 88 responses. All responses were checked manually and 10 of them were incomplete. We didn't consider the incomplete responses for data analysis as a requirement of the questionnaire is not accomplished fully in those 10 incomplete responses. Total 78 responses were evaluated for further processing. The detail of respondents' response is given in section 4.2.

4) SURVEY DATA ANALYSI

The frequency analysis method is used to analyze the qualitative and quantitative data. This approach is effective to measure the ordinal and nominal data between the variables and across the group of variables [44]. This method has been used to statistically compare the identified factors and their significance in the software industry. Several existing studies of other software engineering domain have used the same data analysis approach [41]–[43].

C. PROMETHEE II APPROACH

The adopted approach PROMETHEE (Preference Ranking Organization Method of Enrichment Evaluation) was presented by Brans *et al.* for the first time in 1985 [21] and now several versions of it are available i.e. PROMETHEE I, II, III, IV, V, cluster etc. The application of the approach depends upon the nature of the problem. Considering the advantages this approach is easy to use and have a low level of complexity. Many successful applications have been treated by PROMETHEE approach in various fields such as; banking, chemistry, health care, management, information technology etc. [9], [11], [12], [39]. The PROMETHEE II was selected for this study because it is interactive and is able to classify and order alternatives which are complex and difficult to compare. This approach in addition has some other characteristics like stability, clarity and simplicity [21].

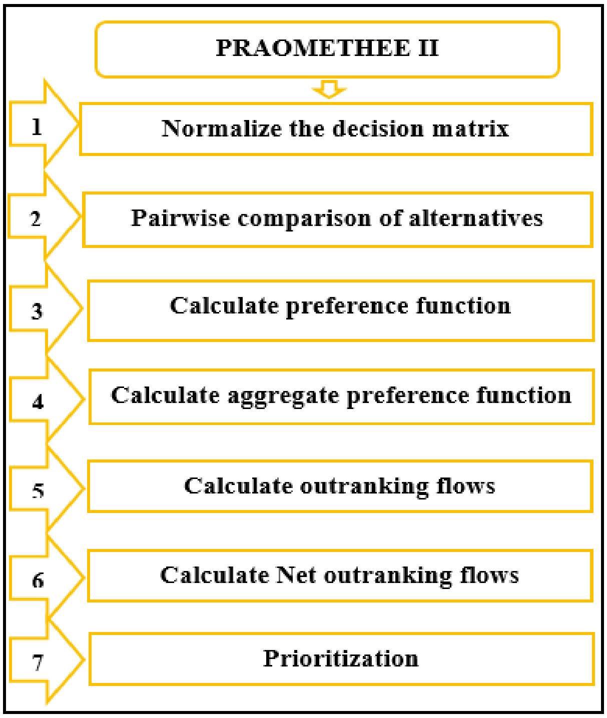

PRAOMETHEE II
1
Normalize the decision matrix
2
Pairwise comparison of alternatives
3
Calculate preference function
4
Calculate aggregate preference function
5
Calculate outranking flows
6
Calculate Net outranking flows
7
Prioritization

FIGURE 4. PROMETHEE II approach steps.

The principles of PROMETHEE II are based on pairwise comparison of alternatives for each criterion (Figure 4). According to Yaghoobi [45] "the alternatives are evaluated according to different criteria which have to be maximized or minimized. Each criterion should be able to distinguish the alternatives, regardless of how the alternatives behave under other criteria". This approach provides complete ranking of alternatives, but as in many multi criteria decision making approaches, decision makers have to identify alternatives by assigning weights, scoring criteria and should have knowledge about out-ranking relationships among different alternatives and the fuzzy variable terms of scale [46].

This method consists of seven steps which are described below:

STEP 1: In this step decision matrix is normalized using beneficial and non-beneficial criteria.

$$R_{ij} = \frac{[D_{ij} - \min D_{ij}]}{[\max(D_{ij}) - \min(D_{ij})]} \quad \text{(beneficial criteria)}$$
$$(i = 1, 2, \ldots, \text{n and } j = 1, 2, \ldots, \text{m}) \qquad (1)$$

$$R_{ij} = \frac{[maxD_{ij} - D_{ij}]}{[\max(D_{ij}) - \min(D_{ij})]} \quad \text{(non-beneficial criteria)} \qquad (2)$$

where D_{ij} is the decision maker's evaluation of the ith alternative with respect to jth criterion.

STEP 2: The difference of ith alternative as compared with other ones are evaluated in this step. This means that the differences in criteria values among various alternatives should be assessed pairwise.

STEP 3: In this step we choose and calculate the preference function, $P_j\ (i, i')$.

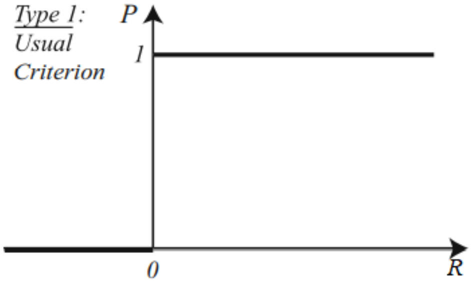


FIGURE 5. Generalized criteria Type 1 preference function.

Mareschal [47] discussed six different types of general preference functions ranges from 0 to 1 (Figure 4). According to Mareschal "the PROMETHEE method induces a preference function to describe the decision maker's preference difference between pairs of alternatives on each criterion". It is possible to choose a different function for each criterion. The usual function (or Type 1) was chosen for this study (Figure 5). Using this function, no parameters and indifference threshold required. This usual criterion function is defined below:

$$P_j(i, i') = 0 \text{ if } R_{ij} \leq R_{i'j} \tag{3}$$

$$P_j(i, i') = 1\ R_{ij} > R_{i'j} \tag{4}$$

STEP 4: The aggregate preference function is calculated by incorporating the weights:

$$\pi\left(i, i'\right) = \sum\nolimits_{j=1}^{m} P_{ij}(i, i')w_j / \sum\nolimits_{j=1}^{m} w_j \tag{5}$$

where w_j is the weight of relative importance given to jth criterion.

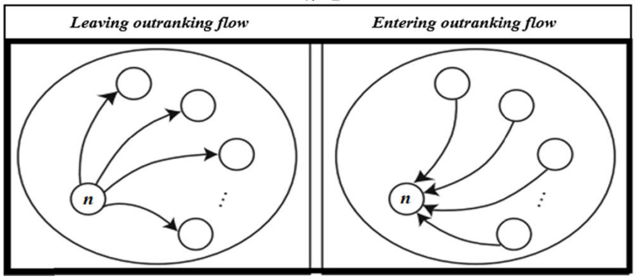


FIGURE 6. PROMETHEE II outranking flows.

STEP 5: For positive and negative outranking flow each alternative is compared with (n-1) alternatives [48], [49] Figure 6. Therefore, it is essential to calculate the entering and leaving outranking flows of each alternative which is given by:

$$\text{The entering flow: } \varphi^{+}(i) = \frac{1}{n-1} \sum\nolimits_{i'-1}^{n} \pi(i', i), \quad (i \neq i') \tag{6}$$

$$\text{The leaving flow: } \varphi^{-}(i) = \frac{1}{n-1} \sum\nolimits_{i'-1}^{n} \pi(i, i'), \quad (i \neq i') \tag{7}$$

where, n represents the number of alternatives. The entering flow measures the weakness of alternatives and leaving flow measures the strength of alternatives.

STEP 6: The PROMETHEE II provides a complete preorder determined by the net outranking flow of decision alternatives:

$$\text{The net flow: } \varphi\ (i) = \varphi^{+}(i) - \varphi^{-}(i) \tag{8}$$

STEP 7: Calculate the ranking of alternatives considering the net outranking flow $\varphi\ (i)$. The highest the $\varphi\ (i)$ the best the alternative. Another advantage of using PROMETHEE II is that it avoids incomparability between alternatives.

IV. STUDY FINDINGS

The section contains the outcomes and analysis of this study.

A. IDENTIFIED CHALLENGES VIA SYSTEMATIC LITERATURE REVIE

Various studies have discussed the factors that could negatively impact the DevOps activities in terms of security. We reviewed those studies briefly and extracted all the challenging factors including *lack of secure coding standards* [2], [23], [80], [81], *using unsuitable performance metrics for security evaluation* [2], [69], [78], [79] and *lack of integrated testing tools to secure DevOps* [14], [52], [53], [71]. Düllmann *et al.* [24] and some other researchers [69], [78], [81] considered *neglecting change control in security* most challenging factor of DevOps. They highlighted that proper security measures among the security and DevOps teams must be shared to make it easy to coordinate and control the security tasks in the distributed software development processing. Rahman and Williams [3] analyzed the selected sets of internet artifacts and surveyed them in nine organizations to explore experts' experiences in DevOps security practices. They explored that *security manual testing and performance measures, threat modeling and scalability, compliance requirements* and *lack of automated testing tools* are the factors having significant impact while securing DevOps activities. The same challenges have been explored in some other researches as well [2], [5], [6], [14], [16], [19], [21], [23], [25], [80]. The software enterprise can only manage the continuous deployment process if they have automated testing tools and performance measures security skills to control such activities. They must have deep knowledge about model of DevOps (CAMS) [25] in order to secure DevOps. They must hire DevOps consultants if they don't have much expertise to manage DevOps security. The consultants will provide step by step guidelines to secure and adopt DevOps activities.

The distributed nature of software development process makes it challenging to secure DevOps activities from various perspectives i.e. processing, communication, deployment, delivery and testing etc [27]. The challenging factor *lack of coordination between security team and DevOps team* is causing more security threats like *untrusted inputs* [54], [55], [74]. The proper implementation of 3C's (communication, coordination, control) must be considered to develop trust and understanding between team members of DevOps. *Developer resistance not to coordinate with DevOps security team* [57], [76], [77] slow-downs the whole deployment process, as no proper measures were taken to control security threats during development phase. There should not be an *unrestricted collaboration between DevOps and security teams* [20], [23], [58], [79] to avoid leakage of information at both ends. The leadership should support their teams to make decisions about the *inadequate channel to monitor team coordination* [52], [53], [71], [79] to control DevOps activities. Proper trainings and meeting sessions should be conducted to improve the expertise of teams. The capabilities to support and encourage team members by leadership to control the *abundance of information causing problems to secure data* [60], [62], [80] will process DevOps smoothly.

The other frequently occurred challenge in an organization to secure DevOps activities is *immature automated deployment tools* [61], [81], [82], due to lack of testing knowledge. The DevOps team must corporate with security team to measure such challenging factors. The enterprise generally *lacks with consistence security polices design and performance measures* [6], [19], [52], [72], [73] due to the rigid policy structure. They should reconsider their policies according to new framework standards for better performance. We have also identified *lack of automated testing tool performance measures for security* [2], [5], [20], [23], [24] as a critical challenge that have a negative impact on DevOps security. Mohan and Othmane [5] surveyed the literature and academia and pointed out that the automated testing tools plays a significance role to integrate security in DevOps. They also investigated that the security and software communities should help industry to develop secure software while applying DevOps activities. The total of 18 challenges were identified and presented in Table 1.

The reported challenges were further divided into four main categories of CAMS model, presented by Edwards and Willis [25]. They classified the DevOps activities into four phases which includes; (Culture, Automation, Measurement and Sharing). This model consists of a set of variables which is useful for the successful implementation of DevOps activities in an organization. We grouped a team consist of three authors for mapping purpose (first three authors). All the team members of the group were using key steps of coding scheme, i.e. code, sub-categories, categories and framework/ theory, of Grounded theory approach [28], [85] as followed by other researchers in their studies [30], [86], [87].

TABLE 1. Identified challenges from literature.

Sr No.	Challenges	Reference
CH1	Lack of automated testing tools	[5],[3],[19],[26],[29]
CH2	Security manual testing and performance configuration	[2],[6],[14],[16],[80]
CH3	Coordination of security team and DevOps team	[5],[15],[16],[19],[26]
CH4	Lack of Automated testing performance measures for security	[2],[5],[20],[23],[24]
CH5	Threat modeling scalability issue	[29],[34],[69]
CH6	Lack of integrated testing tools to secure DevOps	[14],[52],[53],[71]
CH7	Inconsistence security polices design and performance measures	[6],[19],[52],[72],[73]
CH8	Untrusted inputs causing isolation	[54],[55],[74]
CH9	Compliance requirements	[2],[6],[23],[55],[56]
CH10	Developer resistance to integrate security protocols	[57],[76],[77]
CH11	Challenge of unrestricted collaboration	[20],[23],[58],[79]
CH12	Using unsuitable performance metrics for security evaluation	[2],[69],[78],[79]
CH13	Abundance of information is a serious threat to secure data	[60],[62],[80]
CH14	Use of immature automated deployment tools	[61],[81],[82]
CH15	Inadequate channel to monitor the collaboration of teams	[52],[53],[71],[79]
CH16	Lack of secure coding standards	[2],[23],[80],[81]
CH17	Ignorance in static testing for security due to lack of knowledge	[15],[16],[55],[72]
CH18	Neglecting change control in security	[24],[69],[78],[81]

By applying this approach, all the challenges were mapped as shown in Table 2. We labeled identified challenges of security in DevOps as (CH). The aim of this categorization is to perform the PROMETHEE II approach for prioritization of challenges based on values of CAMS principles given in Figure 7. We also verified our finding by applying inter-rater reliability test [68] to remove further biases before applying PROMETHEE II. The three external were invited to verify the mapping scheme results. They counter checked all the steps performed for mapping scheme. After comparing both outcomes we have calculated the Kendalls coefficient of concordance ($W = 0.635$, $p = 0.0005$) shows the significance agreement between both results. The link of code is given in the empirical study section.

TABLE 2. Mapping of identified challenges.

Category	Challenge
Culture	Untrusted inputs causing isolation (CH8)
	Challenge of unrestricted collaboration (CH11)
	Lack of secure coding standards (CH16)
	Neglecting change control in security (CH18)
Automation	Lack of automated testing tools (CH1)
	Threat modeling scalability issue (CH5)
	Lack of integrated testing tools to secure DevOps (CH6)
	Compliance requirements (CH9)
	Use of immature automated deployment tools (CH14)
Measurement	Security manual testing and performance configuration (CH2)
	Lack of Automated testing performance measures for security (CH4)
	Inconsistence security polices design and performance measures (CH7)
	Using unsuitable performance metrics for security evaluation (CH12)
	Inadequate channel to monitor the collaboration of teams (CH15)
Sharing	Coordination of security team and DevOps team (CH3)
	Developer resistance to integrate security protocols (CH10)
	Abundance of information is a serious threat to secure data (CH13)
	Ignorance in static testing for security due to lack of knowledge (CH17)

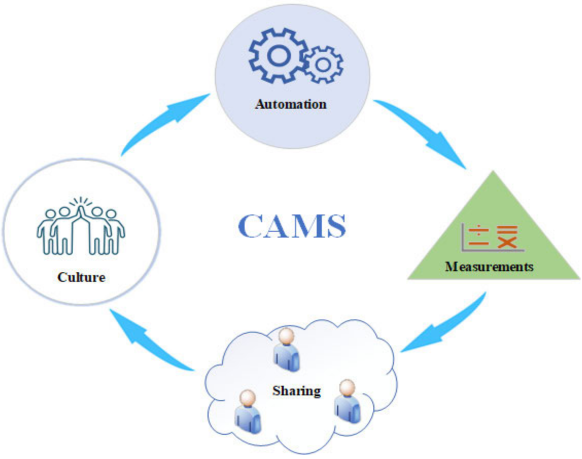


FIGURE 7. DevOps CAMS principles.

Culture (C): The culture is defined as "the interaction of people and groups and is driven by behavior. Substantial communication improvement can result when there is a mutual understanding of others and their goals and responsibilities."

In traditional information technology business development and operation teams works as distinct groups. They spoke different languages as operation staff deals with 'maintaining stable environments and infrastructure' and development team deal with 'innovating, creating and moving fast and breaking things.' These goals without coordination permit both teams to perform their tasks smoothly. By changing this business concept and sharing responsibilities and tasks on the same page is a main goal of DevOps.

Automation (A): Automation deals with many needs and solves various issues of an organization. Automation can save time, cost and effort and is just similar to culture in sharing responsibilities. According to Guthrie [25] "The impact of implementing infrastructure as code as well as using continuous integration and continuous delivery pipelines can be magnified after understanding an organization's culture and goals. It helps to think of automation as an accelerator that enhance the benefits of DevOps as a whole."

Measurement (M): The continuous improvement process is essential while dealing with DevOps environment. Using the measurement will help the practitioners to follow the right direction. DevOps practices will encourage the practitioners to look at the entire system and assess the whole system not just focusing on small sub-sections. The main significance of using measurements has kept a balance on "income, costs, revenue, mean time to recovery, mean time between failures, and employee satisfaction."

Sharing (S): The last phase of CAMS is sharing. Sharing includes three components visibility, transparency, and transfer of knowledge. By sharing the knowledge with teams will help to integrate the organization loop strongly. This collective information will encourage the team to develop a strong unit that can resolve all basic ambiguities which DevOps faces in an enterprise [63].

B. EMPIRICAL STUDY RESULTS

To verify the findings of systematic literature review, a questionnaire survey was conducted with the industrial and academic experts and the analyzed responses of participants are discussed in sub-section below. The questionnaire was designed for survey analysis (Appendix -A). The total of 78 complete responses was collected from participants (DevOps experts) Table 3.

Finstad [40] stated that "the bibliographic data of survey participates give the insight of survey respondents which shows the maturity level of collected data set." Shameem *et al.* [41] underlined that the demographic data is useful to collect information about survey respondents and to verify that the targeted population is related to the particular study. The demographic detail of DevOps experts has been discussed in sub-sections.

1) RESPONDENTS DESIGNATION

Niazi *et al.* [42] underlined that the position of participants in particular organization has influencing impact on the factors. The suggestions of ranking by participants are based on the level of experience the participants have with that factor. Finstad *et al.* [40] defines that the priority of influencing factors varies according to the designation of the targeted population. The designation-based analysis of the respondents

TABLE 3. Responses of survey respondents.

S.#	List of challenges	Number of Responses (N=78)							
		Positive			Negative			Neutral	
		E.A	A	%	D	E.D	%	N	%
C1	**Culture**	**43**	**28**	**91**	**2**	**2**	**5**	**3**	**3**
1	Untrusted inputs causing isolation (CH8)	26	34	78	8	4	15	6	7
2	Challenge of unrestricted collaboration (CH11)	39	30	88	1	2	3	6	7
3	Lack of secure coding standards (CH16)	21	40	88	2	3	6	12	15
4	Neglecting change control in security (CH18)	35	40	96	1	0	1	2	2
C2	**Automation**	**40**	**25**	**83**	**3**	**4**	**8**	**6**	**7**
5	Lack of automated testing tools (CH1)	34	27	78	7	0	8	10	12
6	Threat modeling scalability issue (CH5)	40	23	80	2	1	3	12	15
7	Lack of integrated testing tools to secure DevOps (CH6)	39	28	85	4	0	5	7	8
8	Compliance requirements (CH9)	25	34	75	4	5	11	10	12
9	Use of immature automated deployment tools (CH14)	30	20	64	10	5	19	13	16
C3	**Measurement**	**26**	**39**	**83**	**4**	**2**	**7**	**7**	**8**
10	Security manual testing and performance configuration (CH2)	29	27	71	6	5	14	11	14
11	Lack of Automated testing performance measures for security (CH4)	25	31	71	4	8	15	10	12
12	Inconsistence security polices design and performance measures (CH7)	39	24	80	5	0	6	10	12
13	Using unsuitable performance metrics for security evaluation (CH12)	40	23	80	2	2	5	11	14
14	Inadequate channel to monitor the collaboration of teams (CH15)	30	39	88	0	0	0	9	11
C4	**Sharing**	**33**	**27**	**76**	**6**	**6**	**15**	**6**	**7**
15	Coordination of security team and DevOps team (CH3)	22	36	74	10	4	17	6	7
16	Developer resistance to integrate security protocols (CH10)	31	20	65	7	8	19	12	15
17	Abundance of information is a serious threat to secure data (CH13)	30	41	89	2	0	2	5	6
18	Ignorance in static testing for security due to lack of knowledge (CH17)	26	28	69	8	7	19	9	11

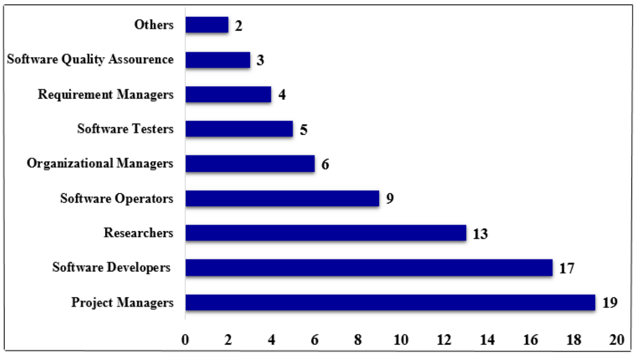


FIGURE 8. Respondents designation- based graph.

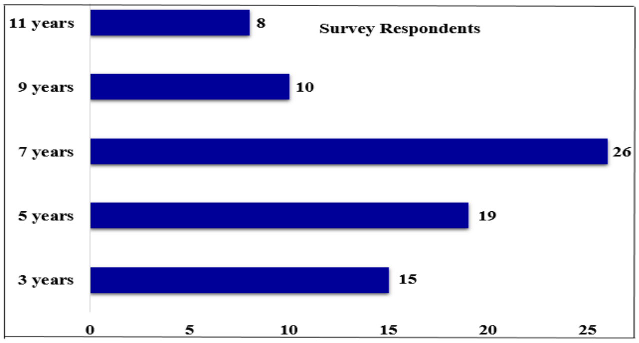


FIGURE 9. Respondents experience.

is presented in Figure 8. The results show that the most of the participant's designations are: researchers, software operators, project managers and software developers.

2) RESPONDENTS EXPERIENCE

The survey participants experience was also analyzed. The medium and mean were calculated and the outcomes shows 5 and 7.5 indicating the young participants respectively. In addition, the significant difference in respondents' experiences were observed briefly. The Figure 9 shows the detail of survey participants graphically.

3) ORGANIZATION SIZE

We collected data related to organization size from the respondent's bibliographic information. By considering the Australian burro of statistics [43] which stated "small (0 to 19 employees), medium (20 to 200 employees) and large (>= 200 employees)". we classified the size of an organization as small, medium and large. Akbar *et al.* [44] highlighted size of an organization as a key entity to assess the maturity level of the survey respondents. The results are graphically explained in Figure 10 showing 40% (small), 30 % (medium), and 30% (large) scale organizations respectively.

4) RESPONSES AGAINST SECURITY CHALLENGES IN DEVOPS

We have collected the responses from industry experts to validate the finding of SLR. The security challenges of DevOps reported in literature were briefly discussed in section 4.1. The responses of survey participants were classified using a

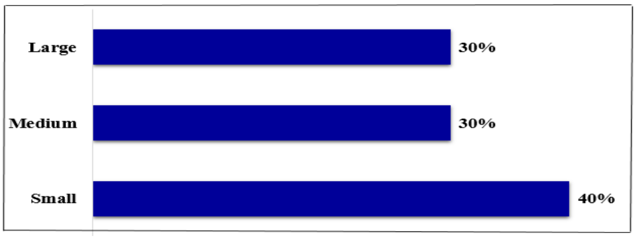


FIGURE 10. Organization size analysis.

Likert scale in positive "extremely agree (EA)", "agree(A)", negative "extremely disagree (ED)", "disagree (D)" and neutral (N). The positive response shows the percentage of those respondents who agreed with the identified security challenges of DevOps. However, the negative feedback shows the frequency of respondents who did not considered the identified challenges have a negative effect on DevOps adoption. The neutral category shows the respondents' feedback who have a neutral opinion about the identified security challenges and their mapping.

The summarized results Table 3 shows that the majority of the survey respondents have a positive opinion about security challenges in DevOps showing the negative effect of factors on DevOps implementation. The frequency analysis shows that percentage of most of the challenges and their categories is > 60%. We have also designed open ended questions in a survey for participants to report additional challenges that were not identified in the available literature study.

However, no additional challenge was marked by the participants. We further noted that **CH18** (neglecting change control in security 96%) is the highest reported challenging factor by survey respondents and "**Culture**" i.e. (C1 = 91%) is the highest category of investigated factors. The same approach has been adopted by other researchers in various fields of software engineering [41], [44].

Hence, based on our findings, we have further prioritize the identified factors by using the PROMETHEE II approach. Each category was compared with all identified challenges in the second survey conducted with DevOps experts. The survey was conducted with the participants who have participated before in the first survey. A total of 25 participants agreed to give a response on second survey Table 4. The questionnaire designed for second survey is provided in Appendix B. The approach discussed in section 3.2 has been used for the assessment of the questionnaire. The sample size is small which may limit our research to some extent. But small sample size has been considered previously in other research studies also [45]–[48]. For example, Wong and Li [49] conducted a survey of the intelligent building systems and evaluate their results based on nine survey responses. Lam and Zhao [50] conducted a survey based on eight participants to evaluate the teaching quality. Shahmeem *et al.* [41] considered five expert's responses to prioritize the factors in agile based organizations. Therefore, based on the above evaluations we can justify 25 as an acceptable sample size for survey analysis. We collected data from 25 practitioners to measure the pairwise comparison of identified challenges using PROMETHEE II approach.

TABLE 4. Percentage ratio of identified challenges (alternatives).

Criterion	C	A	M	S
Weights	0.35	0.25	0.25	0.15
Alternatives Percentage ratio where, n=25				
CH1	69	98	70	59
CH2	70	99	87	67
CH3	88	76	67	75
CH4	100	89	93	82
CH5	98	87	76	65
CH6	56	66	70	52
CH7	73	59	53	64
CH8	81	92	59	85
CH9	53	60	67	70
CH10	97	90	87	82
CH11	86	73	69	76
CH12	100	95	82	89
CH13	79	89	73	66
CH14	82	91	80	58
CH15	90	76	69	79
CH16	98	95	92	89
CH17	55	75	71	68
CH18	89	70	84	71
max value	100	100	93	89
min value	53	59	53	52

To weight the categories of CAMS model, all authors of this research participated. The scale discovered by Bozbura *et al.* [51] was used to convert the linguistic values of weights to numbers. The average weight for each category was calculated which was further used in the PROMETHEE II approach for prioritization.

The weights of categories were further investigated by sending them to three external DevOps experts (one expert from Chongqing University, China and two DevOps experts from Soft-Tech organization Pakistan). After their approval, we have used the weighted categories for further calculation. The Table 4 shows the weighted categories and pairwise percentage ratio of each alternative calculated using a Likert scale explained previously in this section.

C. APPLICATION OF PROMETHEE II APPROACH

To prioritize the identified security challenges of DevOps with categories we used PROMETHEE II approach. The steps followed to perform the application of PROMETHEE II are discussed below:

STEP 1: Normalized the decision matrix by using eq 1and eq 2 is constructed (Table 5) we have selected "automation" as non-beneficial criteria depending upon the nature of automation attributes (i.e. cost, time and effort etc. section 4.1) which should be reduced while developing DevOps environment. All the other criteria are beneficial criteria.

TABLE 5. Normalized decision matrix.

Weights	0.35	0.25	0.25	0.15
Normalized Decision Matrix				
Challenges	C	A	M	S
CH1	0.34	0.05	0.44	0.19
CH2	0.36	0.02	0.85	0.41
CH3	0.74	0.59	0.37	0.62
CH4	1.00	0.27	1.00	0.81
CH5	0.96	0.32	0.59	0.35
CH6	0.06	0.83	0.44	0.00
CH7	0.43	1.00	0.02	0.32
CH8	0.60	0.20	0.17	0.89
CH9	0.00	0.98	0.37	0.49
CH10	0.94	0.24	0.85	0.81
CH11	0.70	0.66	0.41	0.65
CH12	1.00	0.12	0.73	1.00
CH13	0.55	0.27	0.51	0.38
CH14	0.62	0.22	0.68	0.16
CH15	0.79	0.59	0.41	0.73
CH16	0.96	0.12	0.98	1.00
CH17	0.04	0.61	0.46	0.43
CH18	0.77	0.73	0.78	0.51

TABLE 6. Pairwise difference of alternative CH1.

Difference (D)	C	A	M	S
Pairwise Difference CH1				
D(CH1-CH2)	-0.02	0.02	-0.41	-0.22
D(CH1-CH3)	-0.40	-0.54	0.07	-0.43
D(CH1-CH4)	-0.66	-0.22	-0.56	-0.62
D(CH1-CH5)	-0.62	-0.27	-0.15	-0.16
D(CH1-CH6)	0.28	-0.78	0.00	0.19
D(CH1-CH7)	-0.09	-0.95	0.41	-0.14
D(CH1-CH8)	-0.26	-0.15	0.27	-0.70
D(CH1-CH9)	0.34	-0.93	0.07	-0.30
D(CH1-CH10)	-0.60	-0.20	-0.41	-0.62
D(CH1-CH11)	-0.36	-0.61	0.02	-0.46
D(CH1-CH12)	-0.66	-0.07	-0.29	-0.81
D(CH1-CH13)	-0.21	-0.22	-0.07	-0.19
D(CH1-CH14)	-0.28	-0.17	-0.24	0.03
D(CH1-CH15)	-0.45	-0.54	0.02	-0.54
D(CH1-CH16)	-0.62	-0.07	-0.54	-0.81
D(CH1-CH17)	0.30	-0.56	-0.02	-0.24
D(CH1-CH18)	-0.43	-0.68	-0.34	-0.32

STEP 2: To evaluate the difference in criteria values between different alternatives pairwise difference is calculated. For example, pairwise difference of 'lack of automated testing tools' CH1 with respect to other alternatives and criteria is shown in Table 6. The same method has been applied on other alternatives in link https://tinyurl.com/uo29yq2.

STEP 3: To choose and determine the preference function we have used usual function because it does not require any parameter such as preference and indifference thresholds. We replaced the calculated values of pairwise difference considering Eq.3 and Eq.4. For example, the preference function values of "CH1" is shown in Table 7. The same approach has been adopted to calculate the preference function of other challenges in link https://tinyurl.com/uo29yq2.

TABLE 7. Preference function values of CH1.

Preference values (P)	C	A	M	S
Preference Function Values CH1				
P(CH1-CH2)	0.02	0.00	0.41	0.22
P(CH1-CH3)	0.40	0.54	0.00	0.43
P(CH1-CH4)	0.66	0.22	0.56	0.62
P(CH1-CH5)	0.62	0.27	0.15	0.16
P(CH1-CH6)	0.09	0.95	0.00	0.14
P(CH1-CH7)	0.26	0.15	0.00	0.70
P(CH1-CH8)	0.00	0.93	0.00	0.30
P(CH1-CH9)	0.60	0.20	0.41	0.62
P(CH1-CH10)	0.36	0.61	0.00	0.46
P(CH1-CH11)	0.66	0.07	0.29	0.81
P(CH1-CH12)	0.21	0.22	0.07	0.19
P(CH1-CH13)	0.28	0.17	0.24	0.00
P(CH1-CH14)	0.45	0.54	0.00	0.54
P(CH1-CH15)	0.62	0.07	0.54	0.81
P(CH1-CH16)	0.00	0.56	0.02	0.24
P(CH1-CH17)	0.43	0.68	0.34	0.32
P(CH1-CH18)	0.00	0.78	0.00	0.00

TABLE 8. Aggregate preference function of CH1.

Preference values (P)	C	A	M	S
Aggregate Preference Function Values CH1				
P(CH1-CH2)	0.01	0.00	0.10	0.03
P(CH1-CH3)	0.14	0.13	0.00	0.06
P(CH1-CH4)	0.23	0.05	0.14	0.09
P(CH1-CH5)	0.22	0.07	0.04	0.02
P(CH1-CH6)	0.03	0.24	0.00	0.02
P(CH1-CH7)	0.09	0.04	0.00	0.11
P(CH1-CH8)	0.00	0.23	0.00	0.04
P(CH1-CH9)	0.21	0.05	0.10	0.09
P(CH1-CH10)	0.13	0.15	0.00	0.07
P(CH1-CH11)	0.23	0.02	0.07	0.12
P(CH1-CH12)	0.07	0.05	0.02	0.03
P(CH1-CH13)	0.10	0.04	0.06	0.00
P(CH1-CH14)	0.16	0.13	0.00	0.08
P(CH1-CH15)	0.22	0.02	0.13	0.12
P(CH1-CH16)	0.00	0.14	0.01	0.04
P(CH1-CH17)	0.15	0.17	0.09	0.05
P(CH1-CH18)	0.00	0.20	0.00	0.00

According to the ranking of alternatives Figure 10 by PROMETHEE II approach CH1 "*lack of automated testing tools*" is marked as the most critical challenge to secure DevOps implementations. Therefore, there should be proper automation testing tools to monitor the security risks of DevOps. As we are dealing with heterogeneous nature in an organization all sites must coordinate to resolve any ambiguity on time [52], [61]. CH16 "*lack of secure coding standards*" is considered as the challenging factor in terms of DevOps security. Proper security implementations and counter measures must be done to control such risks. The standards of security must be improved with that of other software processing paradigms [4]. Another challenge

TABLE 9. Outranking flows of all alternatives.

	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8	CH9	CH10	CH11	CH12	CH13	CH14	CH15	CH16	CH17	CH18	φ^{+}
CH1	0	0.14	0.34	0.52	0.34	0.29	0.23	0.28	0.45	0.35	0.44	0.18	0.2	0.37	0.49	0.18	0.45	0.2	0.32
CH2	0.01	0	0.31	0.38	0.28	0.27	0.2	0.25	0.32	0.31	0.34	0.13	0.14	0.34	0.35	0.15	0.33	0.2	0.25
CH3	0.02	0.12	0	0.28	0.13	0.1	0.04	0.1	0.22	0.03	0.24	0.04	0.08	0.04	0.28	0.03	0.15	0.08	0.12
CH4	0	0	0.08	0	0.01	0.18	0.01	0.18	0	0.1	0.3	0	0	0.08	0.03	0.09	0.12	0.14	0.08
CH5	0	0.08	0.11	0.19	0	0.17	0.08	0.18	0.14	0.13	0.15	0	0.02	0.12	0.19	0.09	0.18	0.13	0.12
CH6	0.1	0.22	0.24	0.52	0.33	0	0.18	0.11	0.46	0.24	0.48	0.17	0.23	0.28	0.53	0.13	0.34	0.1	0.27
CH7	0.07	0.17	0.2	0.37	0.26	0.2	0	0.24	0.3	0.21	0.3	0.1	0.14	0.23	0.34	0.18	0.35	0.23	0.23
CH8	0.14	0.25	0.28	0.56	0.39	0.16	0.27	0	0.5	0.28	0.52	0.23	0.3	0.32	0.56	0.04	0.38	0.04	0.31
CH9	0	0	0.09	0.07	0.03	0.19	0.01	0.18	0	0.1	0.05	0.01	0	0.09	0.07	0.09	0.12	0.15	0.07
CH10	0.01	0.11	0.01	0.27	0.13	0.09	0.04	0.08	0.22	0	0.24	0.02	0.07	0.04	0.28	0.01	0.13	0.05	0.11
CH11	0	0.03	0.12	0.1	0.05	0.22	0.02	0.21	0.06	0.13	0	0.04	0.02	0.12	0.06	0.12	0.16	0.18	0.10
CH12	0	0.09	0.18	0.34	0.17	0.18	0.09	0.19	0.28	0.19	0.3	0	0.07	0.21	0.35	0.09	0.28	0.14	0.19
CH13	0	0.08	0.21	0.32	0.17	0.22	0.11	0.24	0.26	0.21	0.27	0.04	0	0.24	0.32	0.14	0.26	0.15	0.19
CH14	0.01	0.11	0	0.23	0.1	0.1	0.02	0.1	0.17	0.02	0.19	0.02	0.07	0	0.24	0.02	0.13	0.07	0.09
CH15	0	0	0.12	0.06	0.05	0.22	0.02	0.21	0.03	0.13	0.01	0.04	0.02	0.12	0	0.12	0.15	0.18	0.09
CH16	0.1	0.21	0.27	0.53	0.35	0.23	0.26	0.1	0.47	0.28	0.49	0.19	0.26	0.31	0.53	0	0.38	0.06	0.30
CH17	0	0.02	0.02	0.18	0.07	0.07	0.06	0.06	0.12	0.02	0.15	0	0	0.04	0.19	0	0	0.02	0.06
CH18	0.13	0.27	0.33	0.59	0.4	0.22	0.32	0.11	0.53	0.32	0.55	0.25	0.28	0.26	0.6	0.07	0.14	0	0.33
φ^{-}	0.03	0.11	0.17	0.32	0.19	0.18	0.12	0.17	0.27	0.18	0.30	0.09	0.11	0.19	0.32	0.09	0.25	0.12	0

TABLE 10. Net flow of alternatives.

Challenges	Leaving flow φ^{+}	Enter flow φ^{-}	**Net flow φ**
CH1	0.32	0.03	0.29
CH2	0.25	0.11	0.14
CH3	0.12	0.17	-0.05
CH4	0.08	0.32	-0.24
CH5	0.12	0.19	-0.07
CH6	0.27	0.18	0.09
CH7	0.23	0.12	0.11
CH8	0.31	0.17	0.14
CH9	0.07	0.27	-0.2
CH10	0.11	0.18	-0.07
CH11	0.1	0.3	-0.2
CH12	0.19	0.09	0.1
CH13	0.19	0.11	0.08
CH14	0.09	0.19	-0.1
CH15	0.09	0.32	-0.23
CH16	0.3	0.09	0.21
CH17	0.06	0.25	-0.19
CH18	0.33	0.12	0.21

is CH18 "*neglecting change control in security*" which occurs when DevOps team and security team works with different principles and have obligations to work together. The DevOps teams must be upgraded by providing them platform of training sessions and discussion bench [53]. The DevOps team focus more on coordination of development and operational team, integrating security with DevOps will resolve several issues. The researchers and academic experts agreed to integrate security with DevOps implementations [4], [6]–[8], [59]. The above marked challenges are critical to secure DevOps proper scheduling must be performed to manage such challenges. These steps will help the organization to adopt DevOps without any consideration related to security.

Further, we have calculated the ranking of identified factors,a according to the CAMS model. For validation we send our mapping scheme to three DevOps experts (one expert from Chongqing University China and two experts from Soft-Tech organization Pakistan) section 4.2.4. Based on their recommendations, we rearrange the position of some factors, the final mapping is presented in Table 3. We have followed a formal mapping approach to categories the challenges, applied in other research studies as well [31]–[33], [44]. To avoid uncertainty and vagueness during prioritization we have been introducing a PROMETHEE II approach. The given taxonomy of categories and factors will contribute to enhance the areas which required more security practices for improvement. The overall rank is represented by 'GR' and local rank by 'LR' shown in Figure 12 respectively. Further, we analyzed that "*Security manual testing and performance configuration*" CH 2 ranked 4th in the overall ranking (GR) but ranks 1st (LR) in "*Measurement*" criteria of CAMS model. This ranking of identified challenges of DevOps security will assist the practitioners by considering the significance of factor within the process area and as a whole.

V. DISCUSSION AND SUMMARY

The key objective of this study is to identify and rank the critical challenges that affect the security of DevOps. To address this study objective, we have conducted an SLR study to explore DevOps security challenges available in the literature. The identified challenges were classified further into core categories of CAMS model. For verification of identified

TABLE 11. Ranking of alternatives (challenges).

Sr #	Challenges	Ranking
CH1	Lack of automated testing tools	1
CH2	Security manual testing and performance configuration	4
CH3	Coordination of security team and DevOps team	10
CH4	Lack of Automated testing performance measures for security	18
CH5	Threat modeling scalability issue	11
CH6	Lack of integrated testing tools to secure DevOps	8
CH7	Inconsistence security polices design and performance measures	6
CH8	Untrusted inputs causing isolation	5
CH9	Compliance requirements	15
CH10	Developer resistance to integrate security protocols	12
CH11	Challenge of unrestricted collaboration	16
CH12	Using unsuitable performance metrics for security evaluation	7
CH13	Abundance of information is a serious threat to secure data	9
CH14	Use of immature automated deployment tools	13
CH15	Inadequate channel to monitor the collaboration of teams	17
CH16	Lack of secure coding standards	2
CH17	Ignorance in static testing for security due to lack of knowledge	14
CH18	Neglecting change control in security	3

factors, we have conducted a questionnaire survey study. Finally, we applied PROMETHEE II approach to priorities the investigated factors with respect to their importance in the area of DevOps security.

A. INVESTIGATION OF DEVOPS SECURITY CHALLENGES (RQ1)

To investigate the challenges of DevOps security we have conducted an SLR. The identified factors (Total = 18) present the key areas where team members of DevOps must focus to secure DevOps. The challenges were further classified into four categories according to CAMS model. The section 4.1 shows the identified factors. The factors and their categorization were empirically investigated by questionnaire survey approach.

B. VALIDATION OF CHALLENGES THROUGH QUESTIONNAIRE SURVEY (RQ2)

The questionnaire survey was conducted to identify and validate the factors from industrial practitioners having knowledge about DevOps. The total of 78 responses was collected and the results of the survey show that the identified factors have a negative impact on DevOps security section 4.2.

C. PRIORITIZATION OF DEVOPS SECURITY CHALLENGES (RQ3)

We have used the step by step process of PROMETHEE II approach to prioritize the factors and their categories. The pairwise comparison of each factor with respect to categories were performed. By considering the net outranking flow of each factor we priorities them. The greater the net outranking flow is the highest is the priority of factor with respect to other alternatives. This technique provides better understanding of multi criteria decision making problems by considering the DevOps security challenges and their mapped categories. This technique provides the complete ranking of all alternatives which will assist the DevOps team to focus on areas having security related issues. The results are presented in Table 11. The result show that the **CH1** "*lack of automated testing tools*" is the most critical challenge need to be resolved while dealing with DevOps security in software enterprise. Furthermore, **CH16** "*lack of secure coding standards*" and **CH18** "*neglecting change control in security*" are the second highest ranked challenges while securing DevOps for successful implementation. Figure 11 shows the graphical distribution of challenges based priority measures.

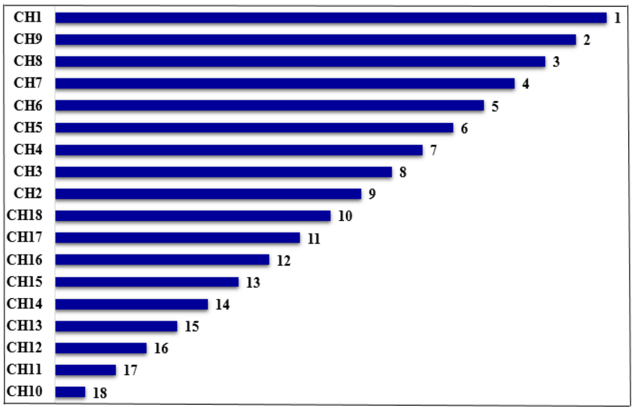


FIGURE 11. Graphical distribution of identified challenges.

D. PRIORITIZATION BASED TAXONOMY OF DEVOPS SECURITY CHALLENGES (RQ4)

The taxonomy of DevOps security challenges was developed based on CAMS model [25]. The core categories of CAMS model are Culture, Automation, Measurement and Sharing briefly discussed in section 4.1. The survey respondents strongly emphasized that all the categories are important for DevOps implementation. They weighted (w) the **Culture** 'w = 0.35' as the highest ranked category as DevOps is all about coordination and commitment of development and operational teams to share their ideas for successful implementation of DevOps. **Automation** 'w = 0.25' and

Measurement ‘w = 0.25’ ranked as the second and third most important categories of investigated DevOps security challenges.

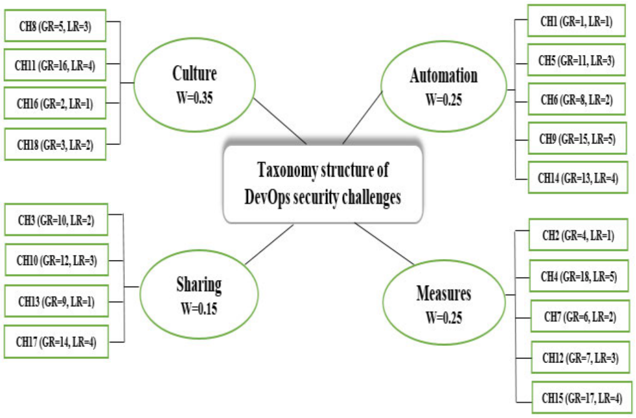


FIGURE 12. Taxonomy structure of DevOps security challenges.

The overall taxonomy of factors with respect to their categories and prioritization is explained in Figure 12. We have investigated that the “*security manual testing and performance configuration*” CH 2 is ranked, i.e. GR = 4 but in **Measurement** category it is ranked, i.e. LR = 1 which shows the significance of this challenge with respect to category. Similarly, “*Abundance of information is a serious threat to secure data*” CH 13 ranked GR = 9 in overall ranking but ranked, LR = 1 in **Sharing** category. The priority of each factor is presented in Figure 12 which will help the practitioners to resolve security concerns of DevOps by applying new practices considering the specific process area.

VI. SUMMARY OF FINDINGS

The main aim of this study is to investigate security related challenges hindering DevOps implementation in software enterprise. We have applied systematic literature study and questionnaire survey approach to identify and validate the factors having a negative impact on secure DevOps adoption. The PROMETHEE II approach has been applied to prioritize the challenging factors concerning to their significance to DevOps security. The brief summary of research question findings is explained in Table 12.

TABLE 12. Findings of research questions.

Research Questions	Findings
RQ1: What are the security challenges of DevOps implementation reported in literature?	The identified challenges are: Lack of automated testing tools, Security manual testing and performance configuration, Coordination of security team and DevOps team, Lack of Automated testing performance measures for security, Threat modeling scalability issue, Lack of integrated testing tools to secure DevOps, Inconsistence security polices design and performance measures, Untrusted inputs causing isolation, Compliance requirements, Developer resistance to integrate security protocols, Challenge of unrestricted collaboration, Using unsuitable performance metrics for security evaluation, Abundance of information is a serious threat to secure data, Use of immature automated deployment tools, Inadequate channel to monitor the collaboration of teams, Lack of secure coding standards, Ignorance in static testing for security due to lack of knowledge, Neglecting change control in security.
RQ2: Are the identified DevOps security challenges related to industry practitioners?	The outcomes calculated after survey shows that the majority of the survey respondent’s shows positive response that the identified challenges fetched from literature study and their classification are important from industrial perspective to assist the implementation of secure DevOps activities.
RQ3: How the identified challenges are prioritized?	Applying PROMETHEE II approach the identified challenging factors and their categories are prioritized. The results show that ‘lack of automated testing tools’ CH1, ‘*lack of secure coding standards*’ CH16, and ‘*neglecting change control in security*’ are highly ranked challenges from security perspective in DevOps implementation.
RQ4: How we can develop the taxonomy for the prioritized investigated challenges?	The ranking of identified challenges as a whole and according to specific criteria Figure 12, will assists the organization practitioners to consider the most critical challenge concerning to their significance in DevOps software enterprise. This taxonomy is based on CAMS model [25,63] by mapping the challenges with each criteria of CAMS model. The authors played a vital role in mapping of factors and further assessment was done by empirical investigation and sending the mapping scheme to three external researchers.

VII. THREATS TO VALIDITY

There are several threats to be addressed towards this study design. For example, there might be a threat to cover all challenges mentioned in literature. This threat may cause incomplete data collection sources, and some relevant studies might be omitted. To reduce this threat, we have developed the search strings to collect data from targeted digital libraries and performed QA steps to verify the quality of research material. The same approach has been adopted by other existing literature review studies [31], [33], [42].

An internal threat to validity is that there might be researcher’s biases while finding literature. To resolve such threat, we performed inter-rater reliability test to check the researcher’s biases, and the results show that the outcomes are consistent and unbiased.

TABLE 13. Survey questionnaire sample to verify the identified devops security challenges.

Survey questionnaire to verify DevOps security challenges				
Part1: Section A (Respondents personal data)				
Name (optional)				
Email address				
Region				
Job title				
Working experience in DevOps related software enterprise				
Current region of your company				
Total work experience in your field				
Have you participated in trainings and workshops activities of DevOps?	No □	Yes □		
Part2: Section A (Respondents enterprise detail)				
Name of enterprise (optional)				
What is primary working of your enterprise?	Research □	Education □	Others □	
	Global software development □	Health □		
Please specify the number of employees in your enterprise?	Less than 20 □	Between 20-100 □	101-200 □	More than 200 □
Please specify the nature of your enterprise?	Multinational □	National □	Not sure □	
Does your enterprise adopted DevOps security activities ?				
How long the DevOps activities being followed in your enterprise?	Years □			
Are the team members in your enterprise concerned about security while implementing DevOps?	No □	Yes □		
Section B: Security challenges in DevOps and categories				
The objective of this section is to specify the factors of DevOps security that have negative influence while adopting DevOps in software enterprise. Please rate the challenge according to the Likert scale based on your experience and understanding.				
Extremely Agree = 'EA', Agree = 'A', Extremely Disagree= 'ED', Disagree= 'D', Neutral= 'N'				

Sr#	Identified Factors and Categories	EA	A	ED	D	N
C1	**Culture**	□	□	□	□	□
1	Untrusted inputs causing isolation (CH8)	□	□	□	□	□
2	Challenge of unrestricted collaboration (CH11)	□	□	□	□	□
3	Lack of secure coding standards (CH16)	□	□	□	□	□
4	Neglecting change control in security (CH18)	□	□	□	□	□
C2	**Automation**	□	□	□	□	□
5	Lack of automated testing tools (CH1)	□	□	□	□	□
6	Threat modeling scalability issue (CH5)	□	□	□	□	□
7	Lack of integrated testing tools to secure DevOps (CH6)	□	□	□	□	□
8	Compliance requirements (CH9)	□	□	□	□	□
9	Use of immature automated deployment tools (CH14)	□	□	□	□	□
C3	**Measurement**	□	□	□	□	□
10	Security manual testing and performance configuration (CH2)	□	□	□	□	□
11	Lack of Automated testing performance measures for security (CH4)	□	□	□	□	□

TABLE 13. ***(Continued.)*** **Survey questionnaire sample to verify the identified devops security challenges.**

12	Inconsistence security polices design and performance measures (CH7)	□	□	□	□	□
13	Using unsuitable performance metrics for security evaluation (CH12)	□	□	□	□	□
14	Inadequate channel to monitor the collaboration of teams (CH15)	□	□	□	□	□
C4	**Sharing**	□	□	□	□	□
15	Coordination of security team and DevOps team (CH3)	□	□	□	□	□
16	Developer resistance to integrate security protocols (CH10)	□	□	□	□	□
17	Abundance of information is a serious threat to secure data (CH13)	□	□	□	□	□
18	Ignorance in static testing for security due to lack of knowledge (CH17)	□	□	□	□	□
Please identify additional challenge: ---						

Another threat to validity is that the sample size n = 78 for questionnaire survey might not be strong enough to justify the validity of challenges. However, based on the other researches in the field of software engineering [30]–[33], [44] this sample size of targeted population is justifiable and for the assessment of factors and their mapping criteria. The response from survey participants were verified to evaluate the significance of identified factors in industrial and academic state.

In this study most of the survey respondents are from Asian countries and we will unable to generalize the outcomes from the other regions perspective. However, the data collected from survey consist of response from different regions and we believe that this sample data is sufficient to validate the factors.

The mapping scheme also has biases to some extent, to validate this concern we have used coding scheme to develop a mapping concept and validate it by empirical study (questionnaire survey) and considering inter-rater reliability test. This shows the significance of categories with respect to mapping alternatives. The same method is adopted in some other studies [30], [86], [87].

There is another threat of ranking the identified and reported challenges, to avoid this threat we prioritize factors through PROMETHEE II approach. The net outranking flow values are used to rank the factors after pairwise comparison of factors with categories, the greatest the outranking value the highest is the rank.

VIII. STUDY IMPLICATIONS

The main focus of this study is to bring in consideration DevOps security challenges due to which practitioners were afraid to implement DevOps activities. This study with detailed systematic literature review and empirical investigation of DevOps security challenges in software organization will provide a body of knowledge to researchers and academic experts, to make strategies and plans for epic secure DevOps adoption.

The identified challenges were mapped into four categories of CAMS model (Culture, Automation, Measurement and Sharing) for further ranking of factors, based on their significance for the successful implementation of secure DevOps environment. These ranks will provide knowledge to the practitioners to consider the most significant factor based on their priorities. The taxonomy of factors will help the practitioners to make improvements in process area which needs further development. By integrating the security practices in DevOps will secure the DevOps environment and provide new thought to validate security measures of DevOps while implementing it.

IX. CONCLUSION AND FUTURE DIRECTIONS

The secure environment of software process development is the first priority of every organization. The software organizations are finding ways to secure their production unit for effective development of products. The DevOps is the most significant approach, which provides more satisfactory results of continuous deployment. The significance of DevOps motivated us to secure DevOps process by exploring the security related challenges faced by practitioners while successful implementation of DevOps.

The SLR was conducted to explore all security related DevOps challenges available in the literature. The identified challenges were further mapped into key categories and verified from industrial and academic experts using questionnaire survey approach. We get 78 total responses which show that the identified factors are related to industrial concerns about DevOps security. These challenges must be resolved for better implementation of DevOps process. Moreover, the PROMETHEE II approach is used for the prioritization of factors and their categories with respect to their importance for the implementation of secure DevOps activities. The ranks were determined using this approach which will assist the practitioners to focus on the key areas which require further improvements to secure DevOps. "*Lack of automated testing tools*" CH1 and "*lack off secure coding standards*" CH16 are the most significant factors and must be addressed first based on their ranking. The taxonomy designed using CAMS model will show the significance of factors and categories for further perfection of DevOps practices. For further analysis, we will investigate the factors that have a positive impact on DevOps security by conducting a literature review and empirical study approach for the successful implementation of DevOps activities.

APPENDIX A

See Table 13.

TABLE 14. PROMETHEE II survey sample.

Survey questionnaire				
Part1: Section A (Respondents personal data)				
Name (optional)				
Email address				
Region				
Job title				
Working experience in DevOps related software enterprise				
Current region of your company				
Total work experience in your field				
Have you participated in trainings and workshops activities of DevOps?	No □	Yes □		
Part2: Section A (Respondents enterprise detail)				
Name of enterprise (optional)				
What is primary working of your enterprise?	Research □	Education □	Others □	
	Global software development □	Health □		
Please specify the number of employees in your enterprise?	Less than 20 □	Between 20-100 □	101-200 □	More than 200 □
Please specify the nature of your enterprise?	Multinational □	National □	Not sure □	
Does your enterprise adopted DevOps security activities ?				
How long the DevOps activities being followed in your enterprise?	Years □			
Are the team members in your enterprise concerned about security while implementing DevOps?	No □	Yes □		
Section B: Security challenges in DevOps and categories				
The objective of this section is to specify the comparison ratio of alternatives and categories which will be used in PROMETHEE II for further pairwise comparison. Please rate the challenge according to the Likert scale based on your experience and understanding.				
Extremely Agree = 'EA', Agree = 'A', Extremely Disagree= 'ED', Disagree= 'D', Neutral= 'N' Where C= Culture, A= Automation, M= Measurement, S= Sharing				

		Categories			
SR#	**Alternatives /Challenges**	**C**	**A**	**M**	**S**
CH1	Lack of automated testing tools				
CH2	Security manual testing and performance configuration				
CH3	Coordination of security team and DevOps team				
CH4	Lack of Automated testing performance measures for security				
CH5	Threat modeling scalability issue				
CH6	Lack of integrated testing tools to secure DevOps				
CH7	Inconsistence security polices design and performance measures				
CH8	Untrusted inputs causing isolation				
CH9	Compliance requirements				
CH10	Developer resistance to integrate security protocols				
CH11	Challenge of unrestricted collaboration				
CH12	Using unsuitable performance metrics for security evaluation				
CH13	Abundance of information is a serious threat to secure data				
CH14	Use of immature automated deployment tools				
CH15	Inadequate channel to monitor the collaboration of teams				
CH16	Lack of secure coding standards				
CH17	Ignorance in static testing for security due to lack of knowledge				
CH18	Neglecting change control in security				

TABLE 15. Quality assessment score of selected studies.

IDs	Reference	QA1	QA2	QA3	QA4	QA5	Total	%age
SP1	[5]	1	0.5	1	1	1	4.5	90
SP2	[3]	0	0.5	0.5	1	1	3	60
SP3	[2]	0	1	1	0	1	3	60
SP4	[6]	1	1	0	1	1	4	80
SP5	[8]	0.5	0.5	1	1	1	4	80
SP6	[14]	1	1	0	1	1	4	80
SP7	[15]	0	1	0.5	0.5	1	3	60
SP8	[16]	1	1	1	0.5	0	3.5	70
SP9	[17]	1	1	0	0	1	3	60
SP10	[19]	0	0.5	0.5	1	1	3	60
SP11	[20]	1	1	1	1	0.5	4.5	90
SP12	[23]	1	0	1	1	1	4	80
SP13	[24]	0.5	0.5	0.5	0.5	1	3	60
SP14	[26]	1	1	0.5	0	0.5	3	60
SP15	[69]	0	0.5	1	1	1	3.5	70
SP16	[29]	1	1	1	1	0	4	80
SP17	[34]	0.5	1	0.5	1	0	3	60
SP18	[52]	1	1	1	0	1	4	80
SP19	[53]	0	1	1	0.5	0.5	3	60
SP20	[70]	1	0.5	1	0	0.5	3.5	70
SP21	[71]	1	0.5	1	0	0.5	3.5	70
SP22	[72]	0	1	1	1	1	4	80
SP23	[54]	1	1	1	1	0.5	4.5	90
SP24	[73]	0	1	1	1	0.5	3.5	70
SP25	[55]	1	0	1	1	1	4	80
SP26	[74]	0	0.5	0.5	1	1	3	60
SP27	[75]	0	1	0.5	0.5	1	3	60
SP28	[56]	1	1	1	1	0	4	80
SP29	[57]	1	0	0	1	1	3	60
SP30	[76]	1	1	0.5	0.5	0	3	60
SP31	[77]	0.5	0.5	1	1	1	4	80
SP32	[58]	1	0.5	1	1	1	4.5	90
SP33	[78]	0	0.5	1	1	1	3.5	70
SP34	[79]	0	0.5	1	1	1	3.5	70
SP35	[80]	1	0.5	1	1	1	4.5	90
SP36	[60]	0	0.5	1	1	1	3.5	70
SP37	[62]	0.5	1	1	1	0	3.5	70
SP38	[61]	1	0	0	1	1	3	60
SP39	[81]	0	0.5	0.5	1	1	3	60
SP40	[82]	1	1	0.5	1	1	4.5	90

Sr#	**QA Questions**	**Checklist Questions**	**Likert scale**
1	QA1	"Does the used research approach address the research questions?"	Yes=1, Partial=0.5, NO=0
2	QA2	"Does the study discuss challenges of DevOps?"	Yes=1, Partial=0.5, NO=0
3	QA3	"Does the study have clear motivation of DevOps implementation?"	Yes=1, Partial=0.5, NO=0
4	QA4	"Is the collected data related to DevOps practices execution?"	Yes=1, Partial=0.5, NO=0
5	QA5	"Are the identified results related to the justification of the research questions?"	Yes=1, Partial=0.5, NO=0

APPENDIX- B

See Table 14.

APPENDIX C

See Table 15.

ACKNOWLEDGMENT

The authors are grateful to the Deanship of Scientific Research, King Saud University for funding through Vice Deanship of Scientific Research Chairs.

REFERENCES

[1] A. Dyck, R. Penners, and H. Lichter, ''Towards definitions for release engineering and DevOps,'' in *Proc. IEEE/ACM 3rd Int. Workshop Release Eng.*, May 2015, p. 3.

[2] B. S. Farroha and D. L. Farroha, ''A framework for managing mission needs, compliance, and trust in the DevOps environment,'' in *Proc. IEEE Mil. Commun. Conf.*, Oct. 2014, pp. 288–293.

[3] A. A. Ur Rahman and L. Williams, ''Security practices in DevOps,'' in *Proc. Symp. Bootcamp Sci. Secur. (HotSos)*, 2016, pp. 109–111.

[4] C. A. Technologies. (2014). *DevOps: The Worst Kept Secret to, Winning in the Application Economy*. Accessed: Jan. 24, 2016. [Online]. Available: http://www. ca.com/us/~/media/Files/whitepapers/devopsthe-worst-kept-secret-to-winning-in-the-applicationeconomy.pdf

[5] V. Mohan and L. B. Othmane, ''SecDevOps: Is it a marketing buzzword?-Mapping research on security in DevOps,'' in *Proc. 11th Int. Conf. Availability, Rel. Secur. (ARES)*, Aug. 2016, pp. 542–547.

[6] A. A. U. Rahman, L. Williams, and M. ay. 2., ''Software security in DevOps: Synthesizing practitioners' perceptions and practices,'' in *Proc. IEEE/ACM Int. Workshop Continuous Softw. Evol. Del. (CSED)*, May 2016, pp. 70–76.

[7] D. G. Feitelson, E. Frachtenberg, and K. L. Beck, ''Development and deployment at facebook,'' *IEEE Internet Comput.*, vol. 17, no. 4, pp. 8–17, Jul. 2013.

[8] X. Bai, M. Li, D. Pei, S. Li, and D. Ye, ''Continuous delivery of personalized assessment and feedback in agile software engineering projects,'' in *Proc. 40th Int. Conf. Softw. Eng. Softw. Eng. Edu. Training (ICSE-SEET)*, 2018.

[9] A. P. U. Siahaan and M. Mesran, ''Fadlina—Best student selection using extended promethee II method,'' *Int. J. Recent Trends Eng. Res.*, Sep. 2017.

[10] P. Liu and Z. Guan, ''Evaluation research on the quality of the railway passenger service based on the linguistic variables and the improved PROMETHEE-II method,'' *J. Comput.*, vol. 4, no. 3, pp. 265–270, Mar. 2009.

[11] I. Veza, S. Celar, and I. Peronja, ''Competences-based comparison and ranking of industrial enterprises using PROMETHEE method,'' *Procedia Eng.*, vol. 100, pp. 445–449, 2015.

[12] S. Theodorou, G. Florides, and S. Tassou, ''The use of multiple criteria decision making methodologies for the promotion of RES through funding schemes in cyprus, a review,'' *Energy Policy*, vol. 38, no. 12, pp. 7783–7792, Dec. 2010.

[13] H. Zhao, Y. Peng, and W. Li, ''Revised PROMETHEE II for improving efficiency in emergency response,'' *Procedia Comput. Sci.*, vol. 17, pp. 181–188, 2013.

[14] A. Balalaie, A. Heydarnoori, and P. Jamshidi, ''Microservices architecture enables DevOps: Migration to a cloud-native architecture,'' *IEEE Softw.*, vol. 33, no. 3, pp. 42–52, May 2016.

[15] D. Cukier, ''DevOps patterns to scale Web applications using cloud services,'' in *Proc. Companion Publication Conf. Syst., Program., Appl., Softw. Humanity (SPLASH)*, 2013, pp. 143–152.

[16] S. Neely and S. Stolt, ''Continuous delivery? Easy! just change everything (well, maybe it is not that easy),'' in *Proc. Agile Conf.*, Aug. 2013, pp. 121–128.

[17] A. A. U. Rahman, E. Helms, L. Williams, and C. Parnin, ''Synthesizing continuous deployment practices used in software development,'' in *Proc. Agile Conf.*, Aug. 2015, pp. 1–10.

[18] S. Cash, V. Jain, L. Jiang, A. Karve, J. Kidambi, M. Lyons, T. Mathews, S. Mullen, M. Mulsow, and N. Patel, ''Managed infrastructure with IBM cloud OpenStack services,'' *IBM J. Res. Develop.*, vol. 60, nos. 2–3, pp. 1–6, Mar./May 2016.

[19] S. de Vries, ''Continuous security testing in a DevOps world,'' in *Proc. OWASP AppSec Eur.* Cambridge, U.K.: Anglia Ruskin Univ., 2014.

[20] J. Smeds, K. Nybom, and I. Porres, ''DevOps: A definition and perceived adoption impediments,'' in *Proc. Int. Conf. Agile Softw. Develop.* Cham, Switzerland: Springer, 2015, pp. 166–177.

[21] V. M. Athawale and S. Chakraborty, ''Facility layout selection using PROMETHEE II method,'' in *Proc. Int. Conf. Ind. Eng. Oper. Manage.*, Bangladesh Dhaka, 2010, pp. 9–10.

[22] B. Fitzgerald and K.-J. Stol, ''Continuous software engineering: A roadmap and agenda,'' *J. Syst. Softw.*, vol. 123, pp. 176–189, Jan. 2017.

[23] J. Humble and J. Molesky, ''Why enterprises must adopt devops to enable continuous delivery,'' *Cutter IT J.*, vol. 24, no. 8, p. 6, 2011.

[24] T. F. Düllmann, C. Paule, and A. van Hoorn, ''Exploiting DevOps practices for dependable and secure continuous delivery pipelines,'' in *Proc. 4th Int. Workshop Rapid Continuous Softw. Eng. (RCoSE)*, May 2018, pp. 27–30.

[25] S. Guthrie. *DevOps Principles—The CAMS Model*. Accessed: May 5, 2019. [Online]. Available: https://medium.com/@seanguthrie/devops-principles-the-cams-model-9687591ca37a

[26] P. Perera, R. Silva, and I. Perera, ''Improve software quality through practicing DevOps,'' in *Proc. 17th Int. Conf. Adv. ICT Emerg. Regions (ICTer)*, Sep. 2017, pp. 1–6.

[27] S. Rafi, W. Yu, and M. A. Akbar, ''Towards a hypothetical framework to secure DevOps adoption: Grounded theory approach,'' in *Proc. Eval. Assessment Softw. Eng.*, 2020, pp. 457–462.

[28] K. J. Stol, P. Ralph, and B. Fitzgerald, ''Grounded theory in software engineering research: A critical review and guidelines,'' in *Proc. IEEE/ACM 38th Int. Conf. Softw. Eng. (ICSE)*, May 2016, pp. 120–131.

[29] P. Frijns, R. Bierwolf, and T. Zijderhand, ''Reframing security in contemporary software development life cycle,'' in *Proc. IEEE Int. Conf. Technol. Manage., Oper. Decisions (ICTMOD)*, Nov. 2018, pp. 230–236.

[30] M. Shameem, A. Khan, and G. Hasan, ''Taxonomy of success factors and their prioritization using analytic hierarchy process for scaling agile process in the global software development,'' *IET Softw.*, Mar. 2020, doi: 10.1049/iet-sen.2019.0196.

[31] A. A. Khan, J. Keung, M. Niazi, S. Hussain, and A. Ahmad, ''Systematic literature review and empirical investigation of barriers to process improvement in global software development: Client–vendor perspective,'' *Inf. Softw. Technol.*, vol. 87, pp. 180–205, Jul. 2017.

[32] M. A. Akbar, M. Shafiq, T. Kamal, M. T. Riaz, and M. K. Shad, ''An empirical study investigation of task allocation process barriers in the context of offshore software development outsourcing: An organization size based analysis,'' *Int. J. Comput. Digit. Syst.*, vol. 8, no. 4, pp. 343–350, 2019.

[33] M. A. Akbar, J. Sang, Nasrullah, A. A. Khan, S. Mahmood, S. F. Qadri, H. Hu, and H. Xiang, ''Success factors influencing requirements change management process in global software development,'' *J. Comput. Lang.*, vol. 51, pp. 112–130, Apr. 2019.

[34] S. Rafi, W. Yu, M. A. Akbar, A. Alsanad, and A. Gumaei, ''Multicriteria based decision making of DevOps data quality assessment challenges using fuzzy TOPSIS,'' *IEEE Access*, vol. 8, pp. 46958–46980, 2020.

[35] M. A. Akbar, J. Sang, A. A. Khan, and S. Hussain, ''Investigation of the requirements change management challenges in the domain of global software development,'' *J. Softw., Evol. Process*, vol. 31, no. 10, Oct. 2019.

[36] M. S. Lewis-Beck, A. Bryman, and T. Liao, ''Semi structured interview,'' in *The SAGE Encyclopedia of Social Science Research Methods*, vol. 1. Newbury Park, CA, USA: Sage, 2003. [Online]. Available: http://books.google.com

[37] P. Singh, A. Pandey, and A. Aggarwal, ''House-to-house survey vs. snowball technique for capturing maternal deaths in India: A search for a cost-effective method,'' *Indian J. Med. Res.*, vol. 125, no. 4, p. 550, 2007.

[38] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian, ''Selecting empirical methods for software engineering research,'' in *Guide to Advanced Empirical Software Engineering*. London, U.K.: Springer, 2008, pp. 285–311.

[39] T. M. Amaral and A. P. C. Costa, ''Improving decision-making and management of hospital resources: An application of the PROMETHEE II method in an emergency department,'' *Oper. Res. Health Care*, vol. 3, no. 1, pp. 1–6, Mar. 2014.

[40] K. Finstad, ''Response interpolation and scale sensitivity: Evidence against 5-point scales,'' *J. Usability Stud.*, vol. 5, no. 3, pp. 104–110, 2010.

[41] M. Shameem, R. R. Kumar, C. Kumar, B. Chandra, and A. A. Khan, ''Prioritizing challenges of agile process in distributed software development environment using analytic hierarchy process,'' *J. Softw., Evol. Process*, vol. 30, no. 11, p. e1979, Nov. 2018.

[42] M. Niazi, S. Mahmood, M. Alshayeb, A. M. Qureshi, K. Faisal, and N. Cerpa, "Toward successful project management in global software development," *Int. J. Project Manage.*, vol. 34, no. 8, pp. 1553–1567, Nov. 2016.

[43] D. Trewin, "Small Business in Australia: 2001," Austral. Bureau Statist., Canberra, NSW, Australia, Tech. Rep. 1321.0, 2002.

[44] M. A. Akbar, J. Sang, A. A. Khan, F.-E. Amin, Nasrullah, S. Hussain, M. K. Sohail, H. Xiang, and B. Cai, "Statistical analysis of the effects of heavyweight and lightweight methodologies on the six-pointed star model," *IEEE Access*, vol. 6, pp. 8066–8079, 2018.

[45] T. Yaghoobi, "Prioritizing key success factors of software projects using fuzzy AHP," *J. Softw., Evol. Process*, vol. 30, no. 1, p. e1891, Jan. 2018.

[46] E. B. Sloane, M. J. Liberatore, R. L. Nydick, W. Luo, and Q. B. Chung, "Clinical engineering technology assessment decision support: A case study using the analytic hierarchy process (AHP)," in *Proc. 2nd Joint 24th Annu. Conf. Annu. Fall Meeting Biomed. Eng. Soc.] [Eng. Med. Biol.*, vol. 3, Oct. 2002, pp. 1950–1951.

[47] L. I. Wen-Ying, "Application of ahp analysis in risk management of engineering projects," *J. Beijing Univ. Chem. Technol. (Social Sci. Ed.)*, vol. 1, pp. 46–48, Feb. 2009.

[48] G. Kabra, A. Ramesh, and K. Arshinder, "Identification and prioritization of coordination barriers in humanitarian supply chain management," *Int. J. Disaster Risk Reduction*, vol. 13, pp. 128–138, Sep. 2015.

[49] J. K. W. Wong and H. Li, "Application of the analytic hierarchy process (AHP) in multi-criteria analysis of the selection of intelligent building systems," *Building Environ.*, vol. 43, no. 1, pp. 108–125, Jan. 2008.

[50] K. Lam and X. Zhao, "An application of quality function deployment to improve the quality of teaching," *Int. J. Qual. Rel. Manage.*, vol. 15, no. 4, pp. 389–413, Jun. 1998.

[51] F. Bozbura, A. Beskese, and C. Kahraman, "Prioritization of human capital measurement indicators using fuzzy AHP," *Expert Syst. Appl.*, vol. 32, no. 4, pp. 1100–1112, May 2007.

[52] M. Senapathi, J. Buchan, and H. Osman, "DevOps capabilities, practices, and challenges: Insights from a case study," in *Proc. 22nd Int. Conf. Eval. Assessment Softw. Eng.*, 2018, pp. 57–67.

[53] L. Zhu, L. Bass, and G. Champlin-Scharff, "DevOps and its practices," *IEEE Softw.*, vol. 33, no. 3, pp. 32–34, May 2016.

[54] D. Trihinas, A. Tryfonos, M. D. Dikaiakos, and G. Pallis, "DevOps as a service: Pushing the boundaries of microservice adoption," *IEEE Internet Comput.*, vol. 22, no. 3, pp. 65–71, May 2018.

[55] V. Mohan, L. ben Othmane, and A. Kres, "BP: Security concerns and best practices for automation of software deployment processes: An industrial case study," in *Proc. IEEE Cybersecur. Develop. (SecDev)*, Sep. 2018, pp. 21–28.

[56] K. Carter, "Francois raynaud on DevSecOps," *IEEE Softw.*, vol. 34, no. 5, pp. 93–96, Sep. 2017.

[57] J. S. Lee, "The DevSecOps and agency theory," in *Proc. IEEE Int. Symp. Softw. Rel. Eng. Workshops (ISSREW)*, Oct. 2018, pp. 243–244.

[58] V. Kaulgud, A. Saxena, S. Podder, V. S. Sharma, and C. Dinakar, "Shifting testing beyond the deployment boundary," in *Proc. Int. Workshop Continuous Softw. Evol. Del. (CSED)*, 2016, pp. 30–33.

[59] H. Myrbakken and R. Colomo-Palacios, "DevSecOps: A multivocal literature review," in *Proc. Int. Conf. Softw. Process Improvement Capability Determination* Cham, Switzerland: Springer, 2017, pp. 17–29.

[60] C. Jansen, "Stabilizing the industrial system: Managed security services' contribution to cyber-peace," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5155–5160, 2017.

[61] H. Yasar and K. Kontostathis, "Where to integrate security practices on DevOps platform," *Int. J. Secure Softw. Eng.*, vol. 7, no. 4, pp. 39–50, Oct. 2016.

[62] M. G. Jaatun, "Software security activities that support incident management in secure DevOps," in *Proc. 13th Int. Conf. Availability, Rel. Secur. (ARES)*, 2018, pp. 1–6.

[63] J. Willis. (2012). *DevOps Culture (Part 1)*. [Online]. Available: https://itrevolution.com/devops-culture-part-1/

[64] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," CiteSeerX Logo, DMCA, Tech. Rep., 2007. [Online]. Available: http://www.elsevier.com

[65] L. Chen, M. A. Babar, and H. Zhang, "Towards an evidence-based understanding of electronic data sources," in *Proc. 14th Int. Conf. Eval. Assessment Softw. Eng. (EASE)*, Apr. 2010.

[66] H. Zhang, M. A. Babar, and P. Tell, "Identifying relevant studies in software engineering," *Inf. Softw. Technol.*, vol. 53, no. 6, pp. 625–637, Jun. 2011.

[67] W. Afzal, R. Torkar, and R. Feldt, "A systematic review of search-based testing for non-functional system properties," *Inf. Softw. Technol.*, vol. 51, no. 6, pp. 957–976, Jun. 2009.

[68] K. A. Hallgren, "Computing inter-rater reliability for observational data: An overview and tutorial," *Tuts. Quant. Methods Psychol.*, vol. 8, no. 1, p. 23, 2012.

[69] M. B. Kamuto and J. J. Langerman, "Factors inhibiting the adoption of DevOps in large organisations: South African context," in *Proc. 2nd IEEE Int. Conf. Recent Trends Electron., Inf. Commun. Technol. (RTEICT)*, May 2017, pp. 48–51.

[70] A. Valani, "Rethinking secure DevOps threat modeling: The need for a dual velocity approach," in *Proc. IEEE Cybersecur. Develop. (SecDev)*, Sep. 2018, p. 136.

[71] E. Rios, E. Iturbe, W. Mallouli, and M. Rak, "Dynamic security assurance in multi-cloud DevOps," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Oct. 2017, pp. 467–475.

[72] L. Williams, "Continuously integrating security," in *Proc. 1st Int. Workshop Secur. Awareness Design Deployment (SEAD)*, 2018, pp. 1–2.

[73] E. Amoroso, "Recent progress in software security," *IEEE Softw.*, vol. 35, no. 2, pp. 11–13, Mar. /Apr. 2018.

[74] S. Faily and C. Iacob, "Design as code: Facilitating collaboration between usability and security engineers using CAIRIS," in *Proc. IEEE 25th Int. Requirements Eng. Conf. Workshops (REW)*, Sep. 2017, pp. 76–82.

[75] T. Quang Thanh, S. Covaci, T. Magedanz, P. Gouvas, and A. Zafeiropoulos, "Embedding security and privacy into the development and operation of cloud applications and services," in *Proc. 17th Int. Telecommun. Netw. Strategy Planning Symp. (Networks)*, Sep. 2016.

[76] M. Shahin, M. Ali Babar, and L. Zhu, "Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices," *IEEE Access*, vol. 5, pp. 3909–3943, 2017.

[77] A. Hudic, M. Flittner, T. Lorunser, P. M. Radl, and R. Bless, "Towards a unified secure cloud service development and deployment life-cycle," in *Proc. 11th Int. Conf. Availability, Rel. Secur. (ARES)*, Aug. 2016, pp. 428–436.

[78] E. Rao, J. Remer, and D. Bauer, "A model for development, transition and technology transfer leading to commercialization of security technology," in *Proc. Int. Carnahan Conf. Secur. Technol. (ICCST)*, Oct. 2014, pp. 1–5.

[79] D. S. Cruzes, M. Felderer, T. D. Oyetoyan, M. Gander, and I. Pekaric, "How is security testing done in agile teams? a cross-case analysis of four software teams," in *Proc. Int. Conf. Agile Softw. Develop.* Cham, Switzerland: Springer, 2017, pp. 201–216.

[80] T. Mackey, "Building open source security into agile application builds," *Netw. Secur.*, vol. 2018, no. 4, pp. 5–8, Apr. 2018.

[81] N. Tomas, J. Li, and H. Huang, "An empirical study on culture, automation, measurement, and sharing of DevSecOps," in *Proc. Int. Conf. Cyber Secur. Protection Digit. Services (Cyber Security)*, Jun. 2019, pp. 1–8.

[82] A. R. Shehab Farhan and G. M. Mostafa Mostafa, "A methodology for enhancing software security during development processes," in *Proc. 21st Saudi Comput. Soc. Nat. Comput. Conf. (NCC)*, Apr. 2018, pp. 1–6.

[83] V. Balali, B. Zahraie, and A. Roozbahani, "A comparison of AHP and PROMETHEE family decision making methods for selection of building structural system," *Amer. J. Civil Eng. Archit.*, vol. 2, no. 5, pp. 149–159, Sep. 2014.

[84] F. Tscheikner-Gratl, P. Egger, W. Rauch, and M. Kleidorfer, "Comparison of multi-criteria decision support methods for integrated rehabilitation prioritization," *Water*, vol. 9, no. 2, p. 68, Jan. 2017.

[85] A. Stauss and J. Corbin, *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. Newbury Park, CA, USA: Sage, 1990, pp. 101–112.

[86] M. A. Akbar, A. A. Khan, A. W. Khan, and S. Mahmood, "Requirement change management challenges in GSD: An analytical hierarchy process approach," *J. Softw., Evol. Process*, to be published.

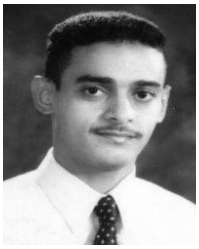
[87] A. A. Khan, M. Shameem, R. R. Kumar, S. Hussain, and X. Yan, "Fuzzy AHP based prioritization and taxonomy of software process improvement success factors in global software development," *Appl. Soft Comput.*, vol. 83, Oct. 2019, Art. no. 105648.

SAIMA RAFI received the M.Sc. degree in computer science from the University of Agriculture Faisalabad, Faisalabad, Pakistan, and the M.S. degree in computer science from Government College University, Faisalabad. She is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology, Chongqing University of Posts and Telecommunications, China. Her research interests include software development and management, DevOps, security concerns in software development, software risk management, and cloud computing security risks. She has an Outstanding Academic Carrier.

WU YU received the bachelor's degree in engineering automation and the Ph.D. degree in automation theory and application from Chongqing University, China, in 1992 and 1997, respectively. Since 1998, she has been working as a Teacher with the Chongqing University of Posts and Telecommunications (CQUPT), China. She is currently a professor.

MUHAMMAD AZEEM AKBAR received the M.Sc. and M.S. degrees in computer science from the University of Agriculture Faisalabad (UAF), Faisalabad, Pakistan, and the Ph.D. degree in software engineering from Chongqing University, China. He is currently working as a Postdoctoral Researcher with the Nanjing University of Aeronautics and Astronautics, Nanjing, China. He has published more than 30 research articles in well-reputed journals and conferences. His research interests include global software development, requirements engineering, empirical studies, global software requirements change management, DevOps implementation, software defect prediction, the Internet of Things, code recommender systems, and software risk management. He has an Outstanding Academic Carrier.

AHMED ALSANAD received the Ph.D. degree in computer science from De Montfort University, U.K., in 2013. He is currently an Associate Professor with the Information System Department and the Chair Member of pervasive and mobile computing with CCIS, King Saud University, Riyadh, Saudi Arabia. He has authored or coauthored more than 12 publications, including refereed IEEE/ACM/Springer journals, conference papers, and book chapters. His research interests include cloud computing, health informatics, ERP and CRM.



ABDU GUMAEI received the B.S. degree in computer science from the Computer Science Department, Al-Mustansiriya University, Baghdad, Iraq, the master's degree in computer science from the Computer Science Department, King Saud University, Riyadh, Saudi Arabia, and the Ph.D. degree in computer science from King Saud University. He is currently an Assistant Professor of computer science. He has worked as a Lecturer and taught many courses, such as programming languages at the Department of Information Systems, King Saud University. He has several researches in the field of image processing. His research interests include software engineering, image processing, computer vision, and machine learning. He has received a patent from the United States Patent and Trademark Office (USPTO), in 2013.