OPEN ACCESS

Kashf Journal of Multidisciplinary Research
Vol: 01 - Issue 12 (2024)
P-ISSN: 3007-1992 E-ISSN: 3007-200X
https://kjmr.com.pk

CHALLENGES AND SOLUTIONS OF CONTINUOUS INTEGRATION AND DEPLOYMENT IN THE DEVOPS CONTEXT: A MULTIVOCAL LITERATURE REVIEW PROTOCOL

Muhammad Ilyas
*Department of Computer Science and IT, University of Malakand Pakistan*milyasmkd@uom.edu.pk
Noor Ul Islam
*Department of Computer Science and IT, University of Malakand Pakistan*noor9877324@gmail.com
Sahab Ahmad Khan
*Department of Computer Science and IT, University of Malakand Pakistan*connectwithsahab@gmail.com
Fazli Rabi
*Department of Computer Science and IT, University of Malakand Pakistan*frfazal100@gmail.com
Nasir Rashid
Department of Computer Science and IT, University of Malakand Pakistan nasir@uom.edu.pk
Corresponding Author: ***Nasir Rashid*** **(**nasir@uom.edu.pk**)**
DOI: *https://doi.org/10.71146/kjmr167*

Article Info

Abstract
CONTEXT- DevOps is a term used to describe the way in which an organization can promote cooperation between the development and operations teams. Its goal is to increase the effectiveness in delivering the products, based on the principles used by the organization. It has to do with trying to achieve cohesive goals, utilizing compatible tools and practices among these teams despite the inclination towards divergence. Joint effort is an important fundamental in the achievement of common goals. Continuous Integration/Continuous Deployment (CI/CD) is a concept of DevOps where the building, testing and deploying of codes is automated.
OBJECTIVES- This research study is intended to explore the CI/CD pipeline challenges/obstacles and solutions/practices in the DevOps context.
METHODOLOGY- We have developed a Multivocal Literature Review (MLR) protocol, and are in the process of implementing the protocol. MLR is based on a structured protocol, and is therefore, different from ordinary literature review. It incorporates both published and gray literature and provides in-depth and more thorough results than ordinary literature review.
EXPECTED OUTCOMES- The findings are expected to assist software vendors to manage the issues involving CI/CD in DevOps projects satisfactorily.

Keywords: *Challenges, continuous integration, continuous deployment, Multivocal Literature Review (MLR).*

Introduction

The *"development and operations"* (DevOps) is a growing strategy to combine the capabilities of development and operational teams and provide enhanced products quickly, reminiscent of the agile methodology [1]. The DevOps is a term that is frequently used to refer to the agile software development technique with the goals of accelerating software development and boosting software quality. The combined duties for software development team and operational team are a fundamental practice that is regularly observed in the DevOps. Agile is an analogy for changing perspective. The DevOps is used to demonstrate cultural values inside the organization. DevOps behave like a framework, whereas Agile is more like an approach[2, 3]. Although, there is no universally accepted definition of DevOps, a number of studies have found several characteristics that DevOps processes often share[2].

Academics and professionals both hold this opinion that the fundamental idea behind DevOps is to eliminate the barriers between development and operations by empowering the DevOps teams with joint accountability and responsibility for all procedures[2]. The developers are more inclined to swiftly offer new characteristics or updates to users, whereas operational team likes to focus on stability, dependability, and security and often suggest that they don't update their products as regularly. Maintaining a frequent release of new software versions can be challenging for the operations team, and such disagreements can hinder software progress [1]. The DevOps serves as a bridge between the operations and development teams, integrating the goals and methodologies of both sides during the development process. In order to address this issue in a constructive manner, it is crucial for both teams to collaborate more effectively throughout the software development process[3].

2. Background

The background of the CI/CD begins with the software development life cycle (SDLC). The SDLC is a multistage, methodical approach to software development. It ensures that high-quality software is produced consistently. The SDLC traces its origins back to the early days of software engineering in the 1960s and 1970s, when computer scientists and engineers sought formal approaches to manage the complexity of software projects [1]. Winston W. Royce introduced one of the earliest SDLC models, the Waterfall Model, in 1970. This linear and sequential approach breaks down the development process into five distinct phases: requirements, design, implementation, verification, and maintenance [1, 4].

In the 1980s, iterative and incremental models emerged to address the constraints of the Waterfall paradigm for major software projects. Iterative and incremental models arose, allowing engineers to improve software by repeating cycles (iterations) and gradually adding functionality (increments). The Spiral Model (1980s), in 1986, Barry Boehm proposed the Spiral model, which combined features of the Waterfall and Iterative models. The Spiral model stresses risk management in iterative cycles that include planning, risk analysis, engineering, and evaluation. Agile Techniques: 2000s Agile approaches sprang to prominence in the early 2000s as an alternative to the rigidity of traditional SDLC models. The Agile Manifesto, released in 2001, emphasized customer collaboration, adaptability, and iterative development [5, 6].

Origins (late 2000s) Patrick Debois popularized the term "DevOps" when he hosted the inaugural DevOps days event in 2009. The movement aimed to bridge the gap between development and operations teams, cultivating a culture of collaboration and shared accountability. The DevOps is a cultural and technical movement that emerged in the late 2000s, emphasizing collaboration between development and operations teams to improve the speed and quality of software delivery. Continuous integration, continuous delivery, automation, deployment, and maintenance are some of the additional DevOps practices that are incorporated into it, building upon the principles of Agile. The CI/CD pipeline practices became crucial to DevOps, allowing for frequent and consistent software releases [7, 8].

The DevOps is the combination of two words, one is Development and other is Operations. It is a culture to promote the development and operation process collectively. This allows a single team to handle the entire application lifecycle, from development to testing, deployment, and operations. DevOps helps you to reduce the disconnection between software developers, quality assurance (QA) engineers, and system administrators et al [2, 9]. The DevOps has a history that dates back to 2007, many frameworks, including the secure continuous deployment framework, are emerging to enable DevOps. The DevOps pipeline automation, Composable DevOps Automated Ontology, and SQUID (Specification Quality in DevOps) framework. In order to effectively understand and adopt DevOps, it is necessary to systematically review and synthesize its approaches given the growing number of DevOps frameworks. DevOps is a set of practices that aims to bridge the gap between developers and operations at its core and covers all the aspects that contribute to high-quality, timely, and optimized software delivery. The Continuous integration and Continuous deployment, log monitoring, automated testing, and updating components are the main focuses of DevOps processes. Although DevOps promises to allow "Faster delivery of builds, features, and bug fixing thereby creating a continuous build pipeline," implementing DevOps is not an easy undertaking [10, 11].

Using a dataset of 84,475 apps from GitHub, GitLab, and Bitbucket, the study investigates the use of CI/CD pipeline services in open-source Android applications. It discovers a poor acceptance rate, with only 10.06% of apps making good use of CI/CD services, of these 59.60% correctly apply CI/CD processes, indicating underutilization. Apps with more GitHub stars indicate that they are typically more popular when they use CI/CD. The most popular tools are Travis, CircleCI, and GitHub Actions. Updates are released twice as frequently in CI/CD-adapted applications, demonstrating improved development efficiency. The article ends with suggestions for increasing the Android developer community's adoption of CI/CD, highlighting advantages like increased project visibility and quicker development cycles [1, 12]. The study uses a mixed-methods approach, starting with interviews to learn what industry professionals think about CI's effect on complexity. Most of them concur that CI increases complexity, but they may accidentally enhance it because of frequent, small modifications that don't undergo enough refactoring. The relationship between cyclomatic complexity in software development and continuous integration procedures is examined in "Big Bangs and Small Pops: On Critical Cyclomatic Complexity and Developer Integration Behavior". It centers on a sizable Java and C/C++ telecom software project with about 200 active developers and frequent commits over a number of years. Here is a quantitative

analysis of commit data spanning 27 months that shows how developer behaviors like commit size and frequency are related to the growth of cyclomatic complexity. The findings indicate that there are considerable differences in developers' continuous integration (CI) practices, which has an impact on complexity evolution. The study highlights how CI can lower the risks that come with high cyclomatic complexity and comes to the conclusion that managing code complexity and guaranteeing software quality requires an understanding of these dynamics [13-15].

3. Research Methodology

The research objective of this work is to carry out MLR methodology to find challenges and solutions of CI/CD pipeline in DevOps context. An MLR refers to assessment of all possible literature that includes formally published academic literature (e.g., journals, conference papers) as well as unpublished and practitioner literature (e.g., white papers, blogposts) to identify, analyze and interpret the phenomena of interest[16] (Garousi, Felderer, & Mäntylä, 2016). MLR studies can give substantial benefits in certain areas of software engineering (SE) in which new developments are occurring and there is a shortage of academic research (Garousi et al., 2016). Multivocal Literature Review (MLR) assessments of all possible literature, including formally published academic literature (e.g., journals, conference papers) as well as unpublished and practitioner literature (e.g., white papers, blog posts) to identify, analyze, and interpret the phenomena of interest[16].

Table 1Published, unpublished and black literature

Published/white literature/ Formal Literature:	**Unpublished/Gray Literature**	**Black Literature**
Articles published in journals, papers presented at conferences, books`	blogs and audio-video (av) media collections of data, electronic prints preprints, presentations, and technical reports.	Tacit knowledge, theories, feelings, ideas, opinions, and thoughts.

It is valuable for SE researchers and practitioners for they offer the reader state of the arts and trends in the field as well as common practices/solutions. Furthermore, the use of grey literature provides an important experience of practitioners in addition to the articles to compensate for the gap between research and practice. The present inclusion of MLR in SE has the potential to contribute to increasing understanding and the completion of systematic reviews with the help of grey literature, which can reveal the entire picture of the subject. Best practices for conducting MLR in SE has been suggested in order to follow the standards that would generate quality results in processes related to MLR. For this purpose, as suggested by Kitchenham and Charters (2007), we have elaborated an MLR protocol to define the plan, which is going to be followed to conduct an MLR study on ‘’challenges and solutions of continuous integration and continuous deployment in DevOps context”.

Identification of the need for review

Before launch the research span, we looked at couple of conference proceeding and related publications to figure out if a Multivocal Literature review on challenges and solutions of the CI/CD pipeline the in DevOps context has been done before. Yet, no Multivocal Literature review of continuous integration and continuous deployment in DevOps was found in the journals and conference proceedings. Consequently, we identified and deemed it appropriate to perform an MLR on the challenges and solutions the CI/CD pipeline the in DevOps context.

Specifying the research questions

The primary objective of this MLR is to discover valuable knowledge regarding the challenges and solutions of the CI/CD pipeline the in DevOps context, in order to bridge the gap in literature and address the challenges and solutions. Therefore, Multi-vocal literature review objectives are:

1) identification of challenges in continuous integration and continuous deployment in DevOps context.
2) Identification of solutions and practice for identified challenges.

We have posed the following two research questions to accomplish the objectives.

RQ1: What are the critical barriers of continuous integration and deployment in DevOps contexts?

RQ2: What are the practices/solution of continuous integration and deployment in DevOps contexts?

Constructing search strategy and search terms

The search strategy is the core of a multivocal literature review and should be well-constructed to retrieve the majority of the studies that will be assessed for eligibility and inclusion. To construct a suitable search term for the research question, the researchers used keywords related to the topic and developed a "search string 1" that included different keywords and their synonyms. The search string was subsequently implemented in various digital literary libraries to retrieve relevant literature. ScienceDirect do not support long search string than eight Booleans, so the “search string 1” was broken down by removing some synonyms and developed a “search string 2 and search string 3” and then executed the derived search string in ScienceDirect library.

The PICO framework is typically used for topics involving an intervention, and it divides the topic into four separate concepts: the patient problem or population, the intervention, the comparison (if there is one), and the outcome(s). using framework like PICO can help identify the search terms needed to search for relevant literature and structure the literature review[6].

Population: Continuous integration and Continuous Deployment in contexts of DevOps

Intervention: Challenges and Solutions

Comparison: this study is comparison less

Outcome of relevance: to find maximum barriers and solution for CI/CD in DevOps contexts.

Synonyms

Barrier: challenges OR barriers OR hurdles OR difficulties OR problems OR inhabited OR limitations

Continuous Integration: Integration, CI, continuous delivery and deployment CD

Deployment: Delivery, installation, implementation

DevOps: Development and operation

Practice: Solution, implementation initiative

Search string using Boolean AND, and OR operators

Constructing search strings

In our MLR, different synonyms of keywords used in RQs are merged with the help of

Boolean operators and OR. Last but not least, searching for other combinations of these keywords, and in this particular research the subsequent search line is defined for each question individually.

Trial search string

Google Scholar and other resource search are quickly scanned using the entered keywords or phrase. In the trial search of the paper some of the identified data sets that are known in the current study are gotten from the above-mentioned databases. If then search string is tried in these types of digital libraries. The naïve trial search actually works like a charm as it goes and retrieved some papers that are well known. We scrutinize the approved search string formulated in the following manner below:

(DevOps OR "Development and operation") AND ("continuous integration" OR "continuous delivery" OR "continuous deployment" OR installation OR implementation OR CI/CD) AND (Barrier OR challenge OR hurdle OR difficult OR problem OR inhabited OR limitation OR Risk OR Factor OR Feature OR Element OR Threat) AND (practice OR Solution OR "implementation initiative" OR Result OR Opportunity OR Technique OR Outcome OR Procedure OR Direction OR Pattern)

String1(S1):

(Barrier OR challenges OR hurdles OR difficult OR problem OR inhabited OR limitation) AND ("continuous integration" OR "continuous delivery" OR CI/CD OR deployment OR Delivery OR installation OR implementation) (DevOps OR "Development and operation")

String2(S2):

("Continuous integration" OR "continuous delivery" OR CI/CD OR deployment OR Delivery OR installation OR implementation) (DevOps OR "Development and operation") AND (practice OR Solution OR "implementation initiative" OR strategy)

String3(S3):

(DevOps OR "development operations") AND (Barrier OR challenges OR hurdles OR problem OR limitation) AND ("continuous integration continuous deployment" OR CI/CD)

String4(S4):

(DevOps OR "development operations") AND (solutions OR practices OR strategies) AND ("continuous integration OR continuous deployment" OR CI/CD)

String5(S5):

(Barrier OR challenge OR hurdle OR problem) AND ("continuous integration continuous deployment" OR CI/CD) AND (DevOps OR "development operations") AND (solutions OR practices OR strategies) AND ("continuous integration OR continuous deployment" OR CI/CD)

Resources to be searched:

In the context of MLR the type of resources that are sought in order to find the related documents are classified as White or Published Literature Resources and GL or Unpublished Literature Resources.

Digital Libraries for search (Published Literature)

According to Kitchenham and Charters (2007), while conducting systematic review of WL in MLR there are two search approaches proposed.

Automatic Search: Automatic search is used to identify the greatest number of documents that have gone through journal peer recognition process from numerous electronic sources. In the present study, the following appropriate online digital sources are used to locate the appropriate literature.

1) **IEEE Xplore (http://ieeexplore.ieee.org/Xplore/home.jsp)**
2) **AIS library (https://aisel.aisnet.org)**
3) **ACM Portal (http://dl.acm.org/)**
4) **Science Direct (www.sciencedirect.com)**
5) **Springer link (http://link.springer.com)**
6) **Wiley Online Library (http://onlinelibrary.wiley.com)**
7) **Scholar Space (http://scholarspace.manoa.hawaii.edu)**

Grey Literature Search (Unpublished Literature)

We found MLR guidelines proposed by Garousi et al[16] in the literature. We used internet search engines like google and yahoo etc. to find documents in the grey literature. The term "grey literature" refers to a wide range of publications, including working papers, government documents, white papers, evaluations, films, and several report formats, including technical, research, project, and annual reports. Usually, commercial publishers do not publish these items and have no control over them. As recommended by Garousi et al[16]. The guidelines recommended using both the Automatic Search and Manual Search techniques. Numerous search engines, such as Google, Thesis Global databases and ProQuest Dissertations, have the ability to perform automated searches. Manual Search methods consist of an informal pre-search. The detail is given below.

Google search engine: An enhanced search will be conducted on a typical Google search engine to obtain suitable materials. thesis, as suggested by Garousi et al[16].

Digital database: An sophisticated search will be steered on the 'ProQuest Dissertations and Thesis Global database to gain the appropriate material on thesis, as Garousi et al[16] directed.

Methods' creator websites: A review must comprise practitioners' websites releasing papers (e.g., Web pages, reports) on the relevant research field, according to MLR criteria given by Garousi et al[16]. An informal pre-search will be carried out to retrieve a list of obtainable sources related to our essential terms to regain the experts' website publishing papers on digital news stories, for this purpose the following links are to be use:

- **Google Search Engine**
- **Social Networks Websites (Facebook, WhatsApp and Instagram)**
- **Relevant blogs**
- **Websites of relevant DevOps (not limited to the following links)**

Stopping criteria

Google search engine opens a lot of links to diverse information sources, but the use of sources should be limited to some number that can be effectively handled. The given search query brings a number of Web sites that are somehow linked to our research study; many of other pages are actually irrelevant. The page rank algorithm needs to be used in order to determine the equitability of the pages that are returned in the search (Langville & Meyer, 2006). It means that a stop is put to searching until a page does not contain any information about CI/CD pipeline in context to DevOps any more.

Primary study selection criteria for published literature

The title, keywords, and abstract of each primary source will be examine in order to

make the initial decision. The goal is to exclude findings that are unrelated to the issue or study topics. The entire texts of the studies will carefully review in order to compare the primary source material collected during the first selection process to the inclusion/exclusion criteria listed. The researchers motivate the research string[17].as well as the recommendations of the authors. To begin, we add the string to the library's metadata. Similar precautions were taken to avoid affecting the title, abstract, and keyword constraints. The first author extracted and documented all articles in a comprehensive manner, taking care of their absolute documentation.

Nevertheless, additional authors analyzed the research in this manner by allocating the connected pieces of details for each study, such as the title's name and an abstract. Based on the aforementioned recommendations, we have developed the following inclusion and exclusion criteria. The researchers encourage more investigation[4]. [Metadata categorization for identifying search patterns in a digital library] as well as the authors' suggestions. We start by adding the string to the library's metadata. A similar effort was made to remember not to compromise the title, abstract, or keyword constraints. The first author carefully collected and meticulously documented each paper, taking care to ensure its complete documentation. However, in doing so, more authors analyzed the research by assigning the relevant parts of each study's details, such as the title and the abstract.

Inclusion criteria (IC)

- IC1: The study is appropriate to the search terms that contain in S1, S2, S3, S4 and S5.
- IC2: The literature that is written only in English language.
- IC3: The study discusses the challenges of CI/CD in DevOps context.
- IC4: The study discusses the solutions and practices of CI/CD in DevOps context.
- IC5: The study period must between 2007 and 2023.

Academic peer-reviewed publications (like journal articles), peer-reviewed experience reports (like conference papers), doctoral and master's theses, reputable consulting firms, and expanding agile structures with online case studies and blog postings as supplementary materials.

Exclusion criteria (EC)

- EC1: Studies that do not emphasis clearly on challenges and solutions of continuous integration and continuous deployment in DevOps context.
- EC2: Studies that do not discuss the continuous integration and continuous deployment in DevOps.
- EC3: Studies whose full-text cannot be accessible.
- EC4: Replica studies (same studies from other journals).
- EC5: Systematic literature reviews (SLRs) or tertiary studies as these studies would reflect matching outcomes in our primary studies.

Study Selection Criteria for Unpublished Literature

Google, ProQuest, and a manual search for relevant sources will be the three sources to which the search string will be applied in the first step. The titles and abstracts will be carefully reviewed using the inclusion and exclusion criteria given in the following sections to remove studies that are not relevant after studies have been gathered through automatic and manual searches. In order to come up with the best solution whenever we run into any problems adding or removing a specific report, we will speak with our research supervisors. The third stage will involve reading the pre-selected studies through to the end and implementing the

inclusion and exclusion criteria to the whole text.

Inclusion Criteria for Grey Literature

- Doctoral and master's theses, dominant consulting firms, and online news outlets with supplementary materials (such as case studies and blog entries) available on their websites.
- The study and the search terms are related.
- The research is composed in the English language.

Exclusion Criteria for Grey Literature:

- Research that doesn't specifically target RQs.
- Research that doesn't discuss continuous integration and deployment in DevOps context.
- Research for which the complete text is not accessible.
- Research articles that are duplicates (the same studies are published in different journals).
- Redundant research items (same studies arranged chronologically, with only the most noteworthy or pertinent study included)

Publication quality assessment for Published Literature

The quality evaluation will be carried out following the publications' final selection. Data extraction and publication quality evaluation will take place sequentially. The following inquiries will be used to evaluate the quality[18].

I. Is it clear how the challenges of continuous integration and deployment in DevOps context identified correctly?

ii. Is it clear how the Solutions for continuous integration and deployment in DevOps context identified correctly?

The subsequent factors (challenges and solutions) will all be indicated with the letters "YES," "NO," "Partial," or "N.A." A small selection will be randomly scored for validation by my supervisor, who is acting as an additional reviewer.

Publication quality assessment for unpublished literature

Grey literature is evaluated using the SADACO (Significance, Authority, Date, Accuracy, Coverage, and Objectivity) checklist. The Fourth International Conference on Grey Literature was held in the month of October 1999 in Washington, DC, USA [19]. Defined grey literature as "that which is produced on all levels of government, academia, business, and industry in print and electronic formats, but which is not controlled by commercial publishers". Written materials that are subjected to scrutiny by subject matter experts during review, such as theses and dissertations, conference papers that are presented by individuals with specialized knowledge or frequently undergo peer review, and a variety of reports written by professionals in the field are all considered to be part of the grey literature. The SADACO method suggests a way to rate the caliber of grey literature. The 19 questions on the SADACO checklist are each given a score of 1, and the source's quality will be deemed acceptable if the total of the scores is greater than 50%; if not, it will be deemed unsuccessful.

- Significance:
- Accuracy:
- Date:
- Authority:
- Coverage:
- Objectivity:

Primary study data extraction

The aim of the current study is to gather data from relevant literature in order to address the research inquiries presented in the review. Each publication will yield information important to the subject under review**.**

- Information about the publication, such as the journal or conference title, authorship, title, and other pertinent information.
- Information or data that is relevant to the current study subject. Table lists the data that must be entered into the data extraction form.

Data extraction is the primary reviewer's responsibility; in the event that problems arise, they will contact the secondary reviewer. An inter-reliability test will be carried out by the primary reviewer following the extraction procedure. In order to compare the results of this test, data extracted by the secondary reviewer at random from various sources will be compared to data extracted by the primary reviewer. The test will be deemed successful if the outcomes are similar. If not, the extracted data will be reviewed once more by the primary reviewer.

Data Storage

A common data collection form works in Microsoft Excel are used to store the extracted data that is, the challenges of the CI/CD pipeline in the DevOps and the solutions/practices while a statistical tool SPSS is used further.

Data synthesis

We have created two summary table for two research questions which are posed in chapter 1, these two tables were made with columns labeled S. No., Challenges, Frequency, and Percentages. These tables presented the list of challenges along with their corresponding frequencies and percentages. A separate table with columns containing the following information will detail each challenge listed in the summary table: challenge group name, S. No. of reference, and paper reference. As was previously mentioned for RQ1, the same practice was also used for RQ2.

4. Preliminary results

One search engine, Google Scholar, and seven digital libraries were used to execute the search strategy. We have used MLR methodology, we originally found 16834 papers, both published and unpublished, to answer two research questions set out in the first chapter. The search strings were then used to retrieve relevant research publications from online digital libraries, databases, and webpages, with inclusion and exclusion criteria. The selection procedure was divided into three stages. After the title and abstract were initially screened, there were 233 articles that met the selection criteria. Following an initial screening of 81 papers, a more extensive Introduction/Discussion selection process was required in the second stage. For this list's final selection in the third stage, the full text read out 43 papers and include 10 from gray literature.

Table 2 List of data collection sources

S.NO	Digital library	Initial Search	Title/Abstract Base Selection	Introduction /Discussion Based Selection	Full Test Based Selection
1	IEEEXPLORE	273	54	20	13
2	AIS ELIBRARY	914	28	9	5
3	SPRINGER LINK	10019	12	2	0
4	WILEY.COM	2621	16	4	3
5	ACM	2182	81	29	16
6	SCHOLARSPACE	602	6	2	1
7	SCIENCE DIRECT	209	11 25	5 10	0 1
8	Google scholar (search engine)	4430	233 ---	81 --	4 10
9	Gray literature	Websites, blogs,			43+14=57
	Total	16820			

5. Validation of result

The final process performed during the conduction of a review is the Validation of the protocol. First, the reached protocol is sent for a secondary review, and last, the protocol is presented to the SE Research Group at the University of Malakand (SERG_UOM). We take suggestions/feedback with regard to the protocol based on which the modifications are made.

6. Conclusion and Future work

While the CI/CD pipeline has gained attention in the DevOps, a MLR is unavailable to address the issues and practices in the applications of CI/CD pipeline in DevOps. It was in this paper at first formulated our plan in terms of an MLR protocol where now is the study in the implementation of this protocol. Furthermore, this study also provides preliminary results. The expected result of following this protocol is to define problems and corresponding approaches/methods of the continuous integration and continuous deployment pipeline of the DevOps with regard to improvement of the productivity and quality on software manufacturing in the software sector.

7. Acknowledgments

We would like to express our gratitude to SERG_UOM for their responses and suggestions regarding this MLR protocol.

References:

[1] G. Bou Ghantous and A. Gill, "DevOps: Concepts, practices, tools, benefits and challenges," *PACIS2017*, 2017.

[2] O. H. Plant, J. van Hillegersberg, and A. Aldea, "Rethinking IT governance: Designing a framework for mitigating risk and fostering internal control in a DevOps environment," *International Journal of Accounting Information Systems*, vol. 45, 2022, doi: 10.1016/j.accinf.2022.100560.

[3] Z. Yiran and L. Yilei, "The challenges and mitigation strategies of using DevOps during software development," ed, 2017.

[4] M. F. Abrar, M. S. Khan, I. Khan, G. Ali, and S. Shah, "Digital Information Credibility: Towards a Set of Guidelines for Quality Assessment of Grey Literature in Multivocal Literature Review," *Applied Sciences*, vol. 13, no. 7, p. 4483, 2023.

[5] L. Leite, C. Rocha, F. Kon, D. Milojicic, and P. Meirelles, "A Survey of DevOps Concepts and Challenges," *ACM Computing Surveys*, vol. 52, no. 6, pp. 1-35, 2020, doi: 10.1145/3359981.

[6] P. Antil, "Requirements engineering in global scaled agile software development environment: a multi-vocal literature review protocol," *arXiv preprint arXiv:2004.12647*, 2020.

[7] N. T. P. GIANG and T. T. M. KHOA, "Automated Continuous Integration Using CircleCI and Firebase for Android Application Development," *Journal of Science and Technology-IUH*, vol. 47, no. 05, 2020.

[8] A. García-Holgado, S. Marcos-Pablos, and F. García-Peñalvo, "Guidelines for performing systematic research projects reviews," 2020.

[9] M. Van Belzen, J. Trienekens, and R. Kusters, "Critical success factors of continuous practices in a DevOps context," 2019.

[10] C. Pang, A. Hindle, and D. Barbosa, "Understanding devops education with grounded theory," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering Education and Training*, 2020, pp. 107-118.

[11] M. Senapathi, J. Buchan, and H. Osman, "DevOps Capabilities, Practices, and Challenges," presented at the Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018, 2018.

[12] P. Liu, X. Sun, Y. Zhao, Y. Liu, J. Grundy, and L. Li, "A first look at CI/CD adoptions in open-source android apps," in *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, 2022, pp. 1-6.

[13] D. Ståhl, A. Martini, and T. Mårtensson, "Big bangs and small pops: on critical cyclomatic complexity and developer integration behavior," in *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, 2019: IEEE, pp. 81-90.

[14] A. Mishra and Z. Otaiwi, "DevOps and software quality: A systematic mapping," *Computer Science Review*, vol. 38, 2020, doi: 10.1016/j.cosrev.2020.100308.

[15] Y. Ska and P. Janani, "A study and analysis of continuous delivery, continuous integration in software development environment," *SSRN Electron. J*, vol. 6, pp. 96-107, 2019.

[16] V. Garousi, M. Felderer, and M. V. Mäntylä, "Guidelines for including grey literature and conducting multivocal literature reviews in software engineering," *Information and software technology*, vol. 106, pp. 101-121, 2019.

[17] T. Bogaard, L. Hollink, J. Wielemaker, J. van Ossenbruggen, and L. Hardman, "Metadata categorization for identifying search patterns in a digital library," *Journal of Documentation*, vol. 75, no. 2, pp. 270-286, 2019, doi: 10.1108/JD-06-

2018-0087.

[18] M. Ilyas and S. U. Khan, "Software integration challenges in global software development environment: A systematic literature review protocol," *IOSR Journal of Computer Engineering (IOSRJCE),* vol. 1, pp. 29-38, 2012.

[19] M. Ashiq, A. Akbar, A. Jabbar, and Q. U. A. Saleem, "Gray literature and academic libraries: How do they access, use, manage, and cope with gray literature," *Serials Review*, vol. 47, no. 3-4, pp. 191-200, 2021.