IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 9 December 2024, accepted 10 February 2025, date of publication 14 February 2025, date of current version 21 February 2025.

*Digital Object Identifier 10.1109/ACCESS.2025.3542630*

SURVEY

# DevOps Integration With Capability Model Maturity Integration: A Systematic Mapping Review

**RANIA RAJAB**[1] **AND MOUHIB ALNOUKARI**[2]
[1]Technology Management Doctorate Program, Syrian Virtual University, Damascus, Syria
[2]Faculty of Engineering, Syrian Private University, Damascus, Syria

Corresponding author: Rania Rajab (rajab.rania@gmail.com)

**ABSTRACT** The Capability Maturity Model Integration (CMMI) is a framework for improving and optimizing organizational processes. DevOps is a revolutionary method that bridges the gap between software development and IT operations by encouraging cooperation and continuous improvement. The combination of CMMI with DevOps is a potential technique for improving software development processes. This systematic mapping review looks at the existing literature on the integration of CMMI and DevOps, identifying how they have been combined, the common aspects between them, the possibility and feasibility of their integration, and whether this integration leads to an improvement in the software development process. A thorough search of numerous databases was done, followed by the application of strict inclusion and exclusion criteria to select relevant studies. A final pool was obtained containing 24 relevant papers. The findings show the benefits of combining CMMI and DevOps to reach a higher level of process improvement, in software development. However, a study about the integration between CMMI and DevOps was not found. Several frameworks have been developed to enhance the combination between them; most of them are models for measuring DevOps maturity based on CMMI. This review emphasizes the need for additional research to develop a model for integrating CMMI with DevOps, which will be the next step in the context of an ongoing study. By giving a comprehensive review of the present state of research, this study adds to a better understanding of how CMMI and DevOps can be effectively coupled to promote software industry advancements.

**INDEX TERMS** CMMI, DevOps, software development process.

## I. INTRODUCTION

Today's software industry is a crucial component for economic advancement, given the challenges developers encounter, including restricted access to data centers, fragmented relationships between developers and operations teams, and increased complexity due to extensive virtualization, in addition to process standardization, quality and risk management, and process maturity.

Many solutions were adopted to tackle these challenges in the context of the software development process. Some of these solutions were traditional, such as Capability Model Maturity Integration (CMMI), which emphasizes the alignment of the process improvement objectives with organizational business objectives by focusing on basic project management, process standardization, quantitative management, and continuous process improvement [1]. Other solutions are based on agile methodologies, from which stemmed DevOps, which embraces capabilities such as continuous delivery, continuous deployment, and continuous improvement; all these accelerate the software improvement process [2]. However, each model has its own challenges, which makes using hybrid models more effective than standalone ones [3]. From this point, much research has been carried out on using DevOps along with a CMMI. Most of the proposed approaches are considered as models measuring the maturity of DevOps using CMMI. No studies discussed the possibility of integration between DevOps and

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

© 2025 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

CMMI, where organizations can achieve a balance between agility and control and create a continuous improvement culture that prioritizes quality, risk management, and process consistency [1], [2]. Customizing CMMI techniques to the unique requirements and dynamics of DevOps settings is critical in order to make sure that the cooperation and automation concepts that are essential to DevOps success are strengthened rather than weakened. Consequently, in order to prove the existence of integration between CMMI and DevOps, it is necessary to analyze the relevant literature. For this purpose, a systematic mapping review was conducted as the research methodology to satisfy the previously indicated needs. This study presents a review that attempts to locate all newly available primary research supporting the integration of CMMI and DevOps in the context of improving the software development process. These efforts were categorized according to the following two factors: the method or module in which CMMI and DevOps are used together, and the commonalities between DevOps and CMMI. The rest of this paper is organized as follows: In Section II, we briefly explain the CMMI product suite and DevOps concepts. Section III provides a quick overview of the related works. Section IV presents the methodology adopted to conduct this study. In Section V, the results were discussed, whereas the findings were described in Section VI. Section VII concludes this research paper.

## II. BACKGROUND

### A. DEVOPS

DevOps was adopted for improving the software industry by producing optimal software production. It is a way of thinking that promotes cross-functional corporation between teams in a software development organization, particularly between development and IT operations, in order to run resilient systems and hasten the delivery of changes [4]. This main goal can be accomplished using several techniques or practices according to certain principles within an environment, which has its own dimensions. These characteristics are summarized and accumulated as follows:

1) **Practices:** They are techniques adopted in building a software product like:
   - **Continuous** (Delivery, Requirement's Engineering, Deployment, Integration, Testing, Monitoring, Experimentation, Learning, Collaboration, Feedback, and Innovation) [5], [6], [7].
   - **Management** (Requirements, Data, Process, Source Code, Automation, Configuration, Traceability, Privileged Access) [2], [8], [9], [10], [11], [12],.
   - **Automated** (Processes, Infrastructure, Testing, Deployment, Monitoring) [2], [6], [13], [14].
   - **Monitoring** (Applications, Resources, Logs, Security, Testing, Quality Assurance, Incident Handling, Actions After Code Commits) [6], [15], [16], [17].
2) **Dimensions:** All terms that address the business areas of the DevOps environment [18]. There is a divergence of views on the DevOps dimensions; some of these views consider dimensions are: Collaboration, Automation, Culture, Monitoring, and Measurement [2], [19], which, according to other views, are DevOps values [14], while other views consider governance, personal traits, departments, and practitioners as DevOps dimensions [2]. Most views are consistent with that dimensions are: processes, tools, culture, and people [14], [17], [20], [21].
3) **Metrics:** Refers to the criteria used in assessing DevOps and measuring the reliability and quality of software product delivery through continuous analysis of strategies, achievements, and services, with internal or external comparison with the best-in-class organizations, and thus implementing self-improvement within or outside the scope of these standards [22]. They are categorized in accordance with DevOps Research and Assessment (DORA) into four metrics that measure software delivery performance and a fifth metric to capture operational capabilities: Deployment Frequency, Lead Time for Changes, Time to Restore Service, Change Failure Rate, and Reliability [23]. According to Ruiz.J. & Others, the previous metrics are not compatible with the Open-Source Software (OSS) projects, so they propose four additional metrics for such kinds of projects as follows: Release Frequency, Lead Time for Released Changes, Time to Repair Code, Bug Issues Rate [21].
4) **Principles:** Are the base of ideals, rules from which DevOps ideology arise. John Willis and Damon Edwards coined the acronym CAMS (Culture, Automation, Measurement and Sharing) which summarizes DevOps principles [24]. Jez Humble later added an L, standing for Lean, to form CALMS [25], but (G. de Franca & others) grouped the DevOps principles in six categories as follows: Social Aspects, Automation, Quality Assurance concerns, Leanness, Sharing and Measurement [26].

### B. CAPABILITY MODEL MATURITY INTEGRATION (CMMI)

CMMI models present a set of best practices for organizations to improve their processes [27]. The CMMI framework does not prescribe how to develop an organization's capabilities; rather, it outlines what should be done in that regard. In addition, when used in conjunction with other industry models and standards, CMMI complements and strengthens performance improvement. Since CMMI V2.0 has been released, CMMI has become a unified model. The various constellations—DEV (development), SVC (services), and ACQ (acquisition)—were replaced by specific capability areas. Currently, CMMI includes multiple domains and views; any organization can select the capability areas, practice areas, domains, or views that are most related to their business operations, and it integrates each domain, like security, into all aspects of the CMMI Categories of "Doing," "Managing," "Enabling," and "Improving" processes. CMMI V3.0 is an integrated CMMI Performance

Solutions ecosystem, consisting of five components: model, appraisal method, training and certifications, systems and tools, and adoption guidelines [28]. Like DevOps, CMMI has its own characteristics as follows:

1) **CMMI V3.0 architecture:** The basic components of the CMMI 3.0 model are: practice areas, views (containing capability areas, categories for capability areas), practice groups, practices, and informative materials. The view is a selected set of model components by the end user or predefined by the CMMI Institute. Predefined views are: development, services, and suppliers. The CMMI V3.0 Model has four category areas, twelve CMMI V2 capability areas, and thirty-one practice areas. Totally, there are eight domains identified to have domain-specific practice areas, namely: data, development, people, safety, security, services, suppliers, and virtual [27].
2) **Dimensions:** A company can improve its business by concentrating on three critical dimensions, according to the Software Engineering Institute (SEI): (1) People with skills, training, and motivation, (2) Procedures and methods defining the relationship of tasks, (3) Tools and equipment [26].
3) **Metrics/Measurements:** The maturity levels are measured by the achievement of the specific and generic goals associated with each predefined set of process areas, where each maturity level matures an important subset of the organization's processes, preparing it to move to the next maturity level [26].
4) **Principles:** The CMMI Institute brings out the guiding principles of CMMI V2.0 as follows: (1) Focus on delivering business value. (2) Focus on business performance and results. (3) Flexible, scalable architecture. (4) Reduce redundancy. (5) Use of plain language. (6) Increased use of graphics and iconography for understanding and adoption [29].

## III. RELATED REVIEWS

Some reviews present DevOps maturity models based on CMMI or compare these models. Eduardo et al. [30] created a reference model-based metrics using a systematic mapping on DevOps Assessment. This study considered 24 primary studies published between 2009 and 2021. Zarour et al. [16] conducted a systematic mapping to identify the primary studies on the methods of implementation, maturity models, and definition of roles in the DevOps framework. This study reviewed 32 relevant primary studies reported between 2015 and 2020. Gasparaite et al. [31] analyzed the availability of DevOps models by means of a systematic literature review. Four papers were selected as relevant between 2009 and 2019. Badshah et al. [32] aimed to develop a DevOps maturity model that can appraise and improve the processes in the DevOps environment. They performed a systematic literature review that resulted in several maturity models and best practices. This study came up with 15 primary studies between 2008 and 2019. Rafi et al. [26] identified 18 critical challenges and associated 73 best practices of DevOps from their systematic literature review, which selected 69 primary studies between 2013 and 2020, as a previous step of proposing a Readiness Model of DevOps (RMDevOps) by adapting CMMI structure, the most significant challenges identified were: resistance to adopting DevOps, a lack of several issues including: effective communication channels, feedback and bug prioritization, awareness of DevOps, maintainability of continuous deployment infrastructure leading to data misuse, integrated testing tools and practices, strategic suggestions from leadership, accountability issues regarding resources, heterogeneity in development and operational structures, entrenched company culture and legacy software, reliance on immature automated deployment tools, risk of disintermediation of roles within teams, DevOps artifacts being tied to specific tools, and the problem of scalability in threat modeling. Jayakody and Wijayanayake [33] applied a systematic literature in purpose of studying the available DevOps maturity models, weaknesses, and strengths of those models. They developed a new framework for DevOps process improvement by monitoring and contrasting 22 primary studies. Akbar et al. [34] developed a DevOps Requirement Change Management and Implementation Maturity Model (DevOps-RCMIMM) by adhering to the CMMI, IMM, and SOPM that could assist the Global Software Development (GSD) organizations in modifying and improving their requirement management process in DevOps. To achieve this, a systematic literature review was contributed to explore 22 success factors and 20 barriers of RCM-DevOps in the GSD context to come up with DevOps-RCMIMM factors by reviewing 107 primary studies between 2000 and 2018. The most prominent success factors are: DevOps change management engineering, a clear change management strategy in DevOps, governance and control of Requirement Change Management (RCM) activities, a vision and goals for change, an advanced and uniform RCM infrastructure at DevOps GSD sites, financial stability in DevOps, motivation of the DevOps RCM team, frequent communication and dialogue in DevOps. The most significant barriers include: requirements tracking and control issues in DevOps, unclear scope of requested changes in DevOps, differing rules and policies across distributed DevOps sites, delays in response from offshore DevOps teams, poor organizational infrastructure in DevOps, and a lack of various elements such as: requirements analysis techniques in DevOps, support from DevOps leadership, resource management across distributed DevOps sites, RCM process training, and change impact analysis in distributed DevOps sites. Zarour et al. [16] identified and benchmarked the DevOps seven maturity models available in the literature. The comparison reveals that most of the maturity models have five stages, either based on CMM or CMMI. The results of their study came up by a simple search in the main databases, which include IEEE, ACM, Springer, and Google Scholar. Plant et al. [35] found that different teams may display the stated DevOps team capabilities

in different ways, so they built an instrument to capture multiple different configurations of a capability rather than just determining whether or not a capability is done to a sufficient degree. After that, a continuous representation is used by the capacity measurement tool to evaluate the individual capabilities on five distinct capability levels. This contrasts with many other maturity models that employ a phased representation whereby maturity levels are assigned to the skills. Automation, lean, and monitoring capability levels were modified from the CMMI continuous representation capability levels. The capabilities and practices that are part of the measurement instrument were based on the results of a systematic literature review that spanned 37 empirical research papers.

## IV. RESEARCH METHOD

This paper follows the systematic mapping review using the guide principles presented in Petersen et al. [36]. This review intends to give an overview of the research area, identify, categorize, and analyze existing literature published between 2010 and 2024, and discuss the possibility of integrating DevOps with CMMI to improve the software development process.

### A. PLANNING

This section outlines the research questions, the research plan, and the criteria for inclusion and exclusion data to refine our findings.

#### 1) RESEARCH QUESTION

We base our work on the following questions:

RQ1: What is the current status of published literature on using CMMI and DevOps together?

RQ2: What are the commonalities between DevOps and CMMI?

RQ3: What are the proposed methods in the literature for using DevOps with CMMI together?

RQ4: Is the integration between DevOps and CMMI possible?

RQ5: Does the integration between DevOps and CMMI lead to improving the software development process?

#### 2) SEARCH STRATEGIES

The search strategy used to locate relevant studies involved conducting automated searches in the databases. The searches were focused on the period from 2010 to May 2024. Systematic mapping was conducted using the following electronic databases: ACM Digital Library, Scopus, IEEE XPlore, Science Direct, and Springer Link. We determined the keywords for the search terms that need to be developed based on the central terms identified in each research question. We looked for studies that were specifically focused on the topic of this research. In this context, Boolean operators like "AND" and "OR" were used to apply the keywords (DevOps, CMMI) from the chosen control articles as guidance for the stage of the exploratory analysis, as explained in Table 1.

**TABLE 1. Used databases and search terms.**

| *Database name* | *Search Term* |
|---|---|
| ACM Digital Library | Title:(DevOps OR CMMI) AND (Abstract:(DevOps AND CMMI) OR Fulltext:(DevOps AND CMMI)) |
| IEEE Digital Library | ("Document Title": DevOps OR "Document Title": CMMI) AND (("Abstract": DevOps AND "Abstract": CMMI) OR ("Full Text Only": DevOps AND "Full Text Only": CMMI)) |
| Scopus | TITLE (DevOps OR CMMI) AND ABS (DevOps AND maturity) |
| Science@Direct | Title, abstract, keywords: "DevOps" AND "CMMI"<br>Title: "DevOps" OR "CMMI" |
| Springer Link | DevOps AND CMMI |

**TABLE 2. Inclusion and exclusion criteria.**

| No | Criteria Name | Inclusion Criteria | Exclusion Criteria |
|---|---|---|---|
| 1 | Content | Studies that address any of the research questions | Studies that didn't tackle using DevOps and CMMI together. |
| 2 | Language | English | Non-English |
| 3 | Availability | Full text is available | Full text is unavailable |
| 4 | Year | Papers published since 2010 | Papers published before 2010 |
| 5 | Research Type | Peer-reviewed journal articles, Conference papers, Book chapters, and Articles written by reputable software development firms, Important university theses. | Secondary research |
| 6 | Duplicated Research | Non- Duplicated | Duplicated |

#### 3) INCLUSION AND EXCLUSION CRITERIA

Inclusion and exclusion criteria were used to select only the relevant studies for answering the defined research questions. These criteria are outlined in Table 2.

### B. CONDUCTING THE SEARCH

There are five primary steps in the review process. These phases are shown in Figure 1, together with the quantity of chosen publications at each level. Starting from the given data sources, we obtained a total of 355 potential papers. In the second step, we used the Parsifal tool[1] to automatically remove duplicate papers, and then we manually applied inclusion and exclusion criteria. There were 277 papers left

[1] https://parsif.al

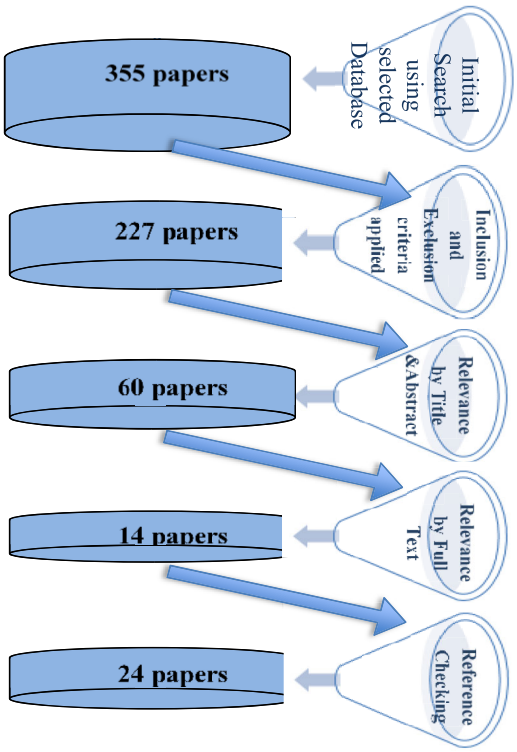



**FIGURE 1.** Search diagram for the paper systematic review.

after this step. The selection criteria based on titles and abstracts are applied to these papers. 60 papers were chosen as a result of the third step. We next went through a full-text assessment of those papers to exclude any that didn't meet our selection criteria. After this point, there are 14 primary research publications left. Eventually, ten more pertinent papers were found after reference checking. Following these phases, there are 24 distinct primary papers included in our final list presented in Table 3.

The distribution of the chosen articles is presented in Table 4.

### C. DATA EXTRACTION AND CLASSIFICATION

To answer the RQs of our research, the required data was collected using a Data Extraction Form (DEF). This form is developed in a table format consisting of the following types of information:

- Bibliometric Information: author(s), publication year, type of publication.
- Research Methodology.
- Aspects related to DevOps, CMMI.
- Proposed methods of using DevOps and CMMI together.
- Results.
- Limitations and constraints.

## V. RESULTS AND DISCUSSIONS

In response to the research questions provided in the proceeding parts, 24 selected papers were analyzed, and the

**TABLE 3.** The final list of selected papers.

| *No* | *Title* | *Author(s)* | *Publication Year* |
|---|---|---|---|
| 1 | Adopting the IBM DevOps approach for continuous software delivery: Adoption paths and the DevOps maturity model | Bahrs, Paul | 2013 |
| 2 | DevOps shifting software engineering strategy Value based perspective | Mohamed, Samer I | 2015 |
| 3 | CMMI Guided Process Improvement for DevOps Projects: An Exploratory Case Study | Rong, Guoping and Zhang, He and Shao, Dong | 2016 |
| 4 | Building Dedicated Project Management Process Basing on Historical Experience | Orłowski, Cezary and Deręgowski, Tomasz and Kurzawski, Miłosz and Ziółkowski, Artur | 2017 |
| 5 | A qualitative study of DevOps usage in practice | F.M. A.Erich, C. Amrit, M. Daneva | 2017 |
| 6 | A capability improvement framework to disrupt, transform and innovate. KPMG International. | KPMG International | 2018 |
| 7 | Adoption Issues in DevOps from the Perspective of Continuous Delivery Pipeline | Toh, M. Zulfahmi and Sahibuddin, Shamsul and Mahrin, Mohd Naz'ri | 2019 |
| 8 | Developing a DevOps maturity model: a validated model to evaluate the maturity of DevOps in organizations | Radstaak, JM | 2019 |
| 9 | Taking DevOps to the Next Level with CMMI V2.0 | CMMI Institute | 2019 |
| 10 | Development of a GQM-based Technique for Assessing DevOps Maturity. | Neubrand, Thomas and Haendler, Thorsten | 2020 |
| 11 | A maturity model for DevOps | Teixeira, Daniel and Pereira, Rúben and Henriques, Telmo and Silva, Miguel Mira Da and João, N A and Faustino, O and Silva, Miguel | 2020 |
| 12 | Capability Counts Series: Reaching CMMI Engineering Practices with DevOps | CMMI Institute | 2020 |
| 13 | CMMI V2.0 contributions and to-dos for organizations | Henriquez, Valeria and A. Calvo-Manzano, Jose and M. Moreno, Ana and San Feliu, Tomas | 2020 |
| 14 | DevOps-RAF: An assessment framework to measure DevOps readiness in software organizations | Marrero, Lilianny and Astudillo, Hernán | 2021 |
| 15 | Readiness model for DevOps implementation in software organizations | Rafi, Saima and Yu, Wu and Akbar, Muhammad Azeem and Mahmood, Sajjad and Alsanad, Ahmed and Gumaei, Abdu | 2021 |

**TABLE 3.** *(Continued.)* **The final list of selected papers.**

| | | | |
|---|---|---|---|
| 16 | DevOps-A Maturity Model for Small Software Companies | Milivojevic, Marko | 2021 |
| 17 | DevOps and organizational performance: The fallacy of chasing maturity | Marnewick, Carl and Langerman, Josef | 2021 |
| 18 | The Alignment between DevOps and Business Outcomes | Mukange, Tsumbedzo and Langerman, Josef | 2022 |
| 19 | A Vision of DevOps Requirements Change Management Standardization | Akbar, Muhammad Azeem and Khan, Arif Ali and Mahmood, Sajjad and Rafi, Saima | 2022 |
| 20 | Design and Validation of a capability Measurement Instrument for DevOps Teams A participatory Action Research Approach | H. Plant, Olivia and Hillegersberg, Jos van and Aldea, Adina | 2022 |
| 21 | DevOps Metrics Objectives and Key Results | Ziaei, Seyedhamidreza | 2022 |
| 22 | Process Improvement Framework for DevOps Adoption in Software Development | Jayakody, J.A.V.M.K. and Wijayanayake, W. M. J. I. | 2023 |
| 23 | DevOps Ontology-An ontology to support the understanding of DevOps in the academy and the software industry | Pardo, César and Orozco, Carlos and Guerrero, Jonathan | 2023 |
| 24 | Investigating effectiveness and compliance to DevOps policies and practices for managing productivity and quality variability | Dan Port and Bill Taber and Parisa Emkani | 2024 |

**TABLE 4.** **The distribution of the selected articles.**

| *Database Name* | *Number of papers* |
|---|---|
| Scopus | 2 |
| IEEE Digital Library | 6 |
| ACM Digital Library | 2 |
| Springer Link | 3 |
| Science@Direct | 1 |
| Others | 10 |
| Total | 24 |

results are described and explained in the following subsections.

### A. (RQ1) WHAT IS THE CURRENT STATUS OF PUBLISHED LITERATURE ON USING CMMI AND DEVOPS TOGETHER?

There is a lake in the published literature that tackles using DevOps and CMMI together. As we mentioned previously, our final list consists of 24 papers. The distribution of these publications by year is presented in Figure 2. This chart illustrates that no contribution was founded before 2013, and 6 papers (24%) have been conducted between 2013 and 2018. Then the interest in combining DevOps and CMMI

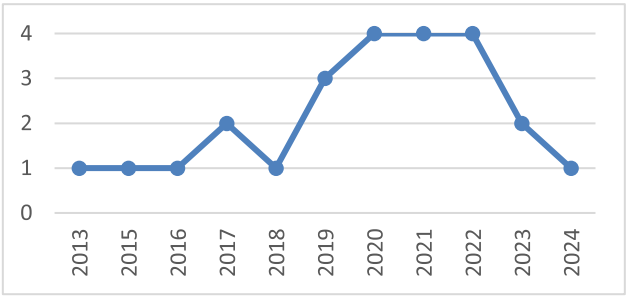



**FIGURE 2.** **Number of published per year.**

was increased between 2019 and 2022, where (64%) have been published, while between 2023 and 2024, three studies (12%) were conducted. The causes of this decrease are likely due to the industry's shifting focus on platform engineering, AI integration, and the development of DevOps models, all of which have contributed to moving research goals away from the precise integration between DevOps and CMMI [59]. Of the chosen studies, 46% were published in journals, 37% were published in conference proceedings, and 17% were white papers.

### B. (RQ2) WHAT ARE THE COMMONALITIES BETWEEN DEVOPS AND CMMI?

There are several concepts shared between DevOps and CMMI (Table 5).

### C. (RQ3) WHAT ARE THE PROPOSED METHODS IN THE LITERATURE FOR USING DEVOPS WITH CMMI TOGETHER?

None of the 24 primary studies resulted from our mapping review tackled the integration between DevOps and CMMI. Some studies presented proposals in the form of models measuring the maturity of DevOps using CMMI or a framework as shown in Table 6:

Previous models have been compared as a preliminary step to develop the integration model of DevOps with CMMI in the context of an ongoing study on this subject as shown in Table 7.

This comparison showed that there is no absolute agreement on the dimensions or focus areas adopted in these models, although most of them considered culture and collaboration as part of these dimensions. Additionally, some of these models were based on identifying critical challenges and successful factors or practices for each maturity level, which ranged between 4 and 6 levels. However, these models generally agreed on the goal of improving the software development process.

### D. (RQ4) IS THE INTEGRATION BETWEEN DEVOPS AND CMMI POSSIBLE?

By reviewing our 24 resulted primary papers, we found that, although DevOps and CMMI are two different approaches in software development and management, each one complements the other within an organization's overall approach.

**TABLE 5.** The commonalities between DevOps and CMMI.

| Concept | Description | Ref. |
|---|---|---|
| Process Improvement | Both CMMI and DevOps emphasize process improvement in software development and IT operations. The CMMI Institute states that CMMI "assists organizations in improving their processes to achieve quantifiable goals," which is consistent with DevOps' emphasis on automation and continuous improvement to accelerate software delivery. | 1, 2, 4, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24 |
| Measurement and Metrics | Metrics and measurement are key tools for driving progress, according to both CMMI and DevOps. The use of metrics such as lead time, deployment frequency, and MTTR (mean time to recovery) to evaluate and enhance delivery processes is emphasized in the DevOps. Metrics are also used by CMMI to track and manage processes to make sure they are operating efficiently and achieving predetermined objectives. | 9, 10, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 24 |
| Continuous Improvement | A continual improvement culture is fostered by both approaches. Teams are encouraged by DevOps to continuously evaluate and improve their processes through iterative improvements, feedback loops, and automation. With the help of CMMI, businesses can adopt and institutionalize process improvements over time, creating a culture of continuous improvement and maturity. | 2, 3, 4, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24 |
| Cross-Functional Collaboration | Both CMMI and DevOps acknowledge the value of cross-functional cooperation within a company. It guarantees that software is delivered effectively and at a high standard, DevOps encourages cooperation between the development, operations, and quality assurance teams. CMMI promotes cooperation amongst the several disciplines that make up the software development lifecycle in order to guarantee that procedures are streamlined and integrated throughout the company | 4, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 20, 22, 23, |
| Quality and Compliance | CMMI offers a framework for creating and overseeing these quality procedures, while DevOps uses automated testing and ongoing input to guarantee quality. Compliance is a topic covered by both CMMI and DevOps, however CMMI focuses on assuring compliance with specified process standards and maturity levels, while DevOps focuses on automating compliance checks as part of the delivery pipeline. | 3, 7, 9, 10, 11, 13, 14, 15, 16, 17, 18, 19, 22, 24 |
| Data-driven decision making | Both the DevOps and CMMI frameworks rely largely on data-driven decision making, with DevOps primarily utilizing metrics, continuous feedback, and operational insights. In order to evaluate the maturity and efficacy of processes, CMMI assessment methods involve data collection and analysis. They also promote the use of data to assist decision-making at all organizational levels. | 3, 4, 7, 9, 10, 12, 14, 17, 19, 23, 24 |
| Customer-Centric Approach | In both CMMI and DevOps, process success and alignment with customer needs depend on the adoption of a customer- centric strategy. DevOps emphasizes Continuous Feedback Loop and promotes collaboration between development, operations, and other stakeholders. It uses Automation which can streamline customer interactions, and in addition to that Continuous monitoring and analytics help in understanding how customers are using the product. CMMI focuses on understanding and managing customer requirements throughout the development lifecycle. It emphasizes measuring and improving customer satisfaction, and incorporates customer feedback into the improvement process, and encourages clear communication and commitment to customer needs at all levels of the organization. | 3, 7, 9, 11, 12, 13, 15, 16, 19, 20, 22, 23, 24 |
| Risk management | Risk Management is an essential component of both DevOps and CMMI. In DevOps it is integrated into the CI/CD pipeline, DevOps adopts Automation of Risk Assessment, Culture of Responsibility, Resilience Engineering and Feedback and Iteration. CMMI includes specific process areas dedicated to risk management, such as "Risk Management" (RSKM) and "Decision Analysis and Resolution" (DAR). These process areas define systematic approaches to identify, analyze, and mitigate risks throughout the project lifecycle. CMMI emphasizes Risk Identification and Analysis, Risk Mitigation Planning, Monitoring and Control, Documentation and Traceability of risk management activities, including risk registers, mitigation plans, and lessons learned. | 3, 4, 7, 9, 12, 13, 17, 18, 19, 21, 23 |
| Customer satisfaction | Customer satisfaction is a crucial aspect in both DevOps and CMMI, with some differences in approach and emphasis. DevOps promotes Continuous Feedback, Customer-Centric Culture, Faster Delivery, Quality Assurance and Service Reliability. CMMI includes specific process areas such as "Customer Satisfaction", and CMMI emphasizes Customer-Focused Processes, Process Maturity, Risk Management and Mitigation and Stakeholder Engagement which approach fosters trust and satisfaction. | 3, 4, 7, 11, 12, 14, 15, 16, 17, 19, 20, 22, 23, 24 |
| Operational efficiency | It is a critical goal in both DevOps and CMMI although they approach it from different point of view due to their respective focuses on agility and process maturity. In DevOps through emphasizing Automation, CI/CD, Scalability and Flexibility, Monitoring and Analytics and Collaboration and Communication. In CMMI through Process Standardization, Process Automation which not as automation-centric as DevOps otherwise it targets areas like requirements management, configuration management, and quality assurance processes, Metrics-Driven Approach, Risk Management and Continuous Improvement. | 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 |

**TABLE 5.** *(Continued.)* **The commonalities between DevOps and CMMI.**

| | | |
|---|---|---|
| Software Testing | Software testing in DevOps and CMMI frameworks focuses on ensuring quality and reliability. DevOps integrates testing early and continuously throughout the CI/CD pipeline, Automation, Shift-left Testing, Continuous Feedback and Testing Tools. CMMI mandates structured testing processes as part of its managed and defined process areas, Traceability, Quality Assurance Activities, Test Management, Verification and Validation. | 3, 4, 7, 9, 12, 13, 14, 15, 16, 19, 23, 24 |
| Broad systems approach | Broad systems approach is a comprehensive view point that, while tackling issues or putting improvements into place, take into account the company as a whole as well as all of its interconnected components. This methodology relates to CMMI and DevOps in the following ways: DevOps adopts End-to-End Automation, Culture and Collaboration, Continuous Improvement, Resilience Engineering and Customer-Centricity. CMMI promotes a systematic and integrated approach, Organizational Alignment: CMMI focuses on aligning organizational goals and processes to achieve business objectives, Risk Management, Performance Management and Continuous Process Improvement. | 2, 3, 4, 7, 9, 10, 12, 14, 17, 18, 21, 23 |

GroBe - Rhode [52] highlighted the methodology integration and found that, when two artifacts have to be integrated, it has to compare models, stat correspondences between them, and check their consistency. By taking [53] into consideration, we present the following points:

(1) Commonalities between DevOps and CMMI, which were discussed previously in the RQ2 answer.
(2) Using DevOps and CMMI together in proposed methods highlighted in RQ3.
(3) Researchers' viewpoints that emphasized the integration between CMMI and Agile from which DevOps stemmed [51], [53], [54], [55], [56].
(4) Industrial reports that showed the incremental use of DevOps and CMMI together in software development companies [29], [57].
(5) Within CMMI, there are two types of informative material: ''explanatory'' and ''context-specific.'' The context-specific section contains guidelines clarifying the meaning of PAs or practices in particular contexts. Some agile contexts mentioned by the CMMI Institute are: Agile with Scrum, DevOps, and others [28], [58].
(6) The CMMI Institute highlighted the need for an integrated approach that enables software companies to coordinate the development and delivery of a blend of software and related services presented in the combination between DevOps and CMMI. The CMMI institute also emphasized that CMMI ensures the success

**TABLE 6.** **The proposed methods in the literature for using DevOps with CMMI together.**

| Proposal | Description | Ref. |
|---|---|---|
| Teixeira maturity model | This model was built according of CMMI MM to enable assessing any organization working model/state against DevOps model. It developed grounded on scientific and professional viewpoints and demonstrated in a real organization. | 11 |
| Maturity model | This model was based on CMMI and adapted to fit a DevOps maturity assessment and should be able to be adapted easily inside organizations and assist them in:<br>1- reaching the necessary maturity level.<br>2- Showing other members of the organization what is required for a successful DevOps environment. | 8 |
| Readiness model RMDevOps | Readiness model for DevOps implementation in software by adapting concept of SOVRM, Rise RM, CMMI, and software process improvement readiness model (SPIRM), it consists of six levels, each one contains various challenges to be overcome with defined practices. These changes can enhance the capability and usability of RMDevOps in the software industry. The assessment of model has been done by using the Motorola assessment tool. | 15 |
| Samer I. Mohamed maturity model | This model depends on the five-levels CMMI maturity model. Each level was evaluated based on four criteria: governance, communication/collaboration, automation, and quality. The organization must strengthen the four dimensions in order to advance from one maturity level to the next. This suggested model, which covers the full life cycle of the application or given service, is derived from the HP model. | 17 |
| IBM DevOps maturity model | This model is meant to be consistent with the goals and results of other specialized models like Agile, SOA, and CMMI. It offers a means to evaluate and view maturity throughout the DevOps continuum. | 1 |
| Capability Measurement Instrument for DevOps Teams | It is an instrument encompasses five dimensions, covering seventeen capabilities and thirty-eight associated practices to capture multiple different configurations of a capability. Automation, lean, and monitoring capability levels were modified from the CMMI continuous representation capability levels. | 20 |

**TABLE 6.** *(Continued.)* **The proposed methods in the literature for using DevOps with CMMI together.**

| | | |
|---|---|---|
| DevOps-RAF | DevOps-RAF takes into consideration the maturity models based on CMMI, so it includes a continuous improvement process and the analysis of gaps between the current level and the level desired by the organization. In addition, DevOps-RAF includes metrics that had not been previously analyzed, like a measurement plan to close the gaps between different skill levels. It was used to (1) evaluate small and medium enterprises' (SME readiness to adapt and adopt DevOps practices, culture, and tooling, (2) identify capability gaps, and (3) propose improvement roadmaps. | 14 |
| Framework for DevOps process improvement | It consists of four main DevOps success areas. Each area follows five maturity levels starting with beginning to expert. It is proposed grounded on evaluation of the data available from existing studies on the DevOps maturity models most of them based on CMMI by comparing the strengths and weaknesses of each model and it is validated by industry practitioners. This framework assists software development companies in obtaining benefits while reducing the difficulties associated with DevOps adoption. | 22 |
| DevOps requirement change management and implementation maturity model (DevOps-RCMIMM) | Based on the existing DevOps and RCM literature by adhering to the CMMI structure. The twenty-five process areas (PAs) that make up the CMMI levels have comprised a variety of real-world approaches. Five maturity levels are identified for the PAs. This model will help practitioners assess and raise the DevOps organization's maturity level. | 19 |
| Dedicated Project Management Process | It is a hybrid process based on the CMMI for Development Model. It incorporates parts of both Agile and Waterfall methodologies, as well as many engineering practices and techniques for process improvement. This process is an important part of Agile Transformation. | 4 |

of DevOps projects; as mentioned, ''A time-tested set of global best practices for continually improving performance, CMMI is built on many of the same principles that underlie the DevOps approach, including a broad systems approach, engaging stakeholders throughout the product lifecycle, and continuous improvement. It's also an integrated model that supports software, hardware, and systems development—including the practices recommended by the DoD—as well as service delivery'' [20], [21].

The previous discussion leads to the fact that integration between DevOps and CMMI is possible.

### E. (RQ5) DOES THE INTEGRATION BETWEEN DEVOPS AND CMMI LEAD TO IMPROVE SOFTWARE DEVELOPMENT PROCESS?

All 24 resulted primary studies emphasized that using DevOps and CMMI together improves the software development process. Depending on the answer of RQ4, the integration between them is a strategic method for software development improvement. While CMMI focuses on process improvement and maturity, DevOps emphasizes cooperation, automation, and quick deployment. Organizations may improve process efficiency, quality assurance, and continuous improvement in addition to achieving faster and more dependable software delivery by merging these frameworks. Using the structured methodology of CMMI, this integration ensures solid, well-defined processes that foster sustainable growth and innovation while utilizing the characteristics of DevOps in agile development settings.

## VI. MAIN FINDINGS

This section summarizes the main conclusions and findings as follows:

(1) No study has yet addressed clearly the integration between DevOps and CMMI, in addition to the scarcity of studies that have presented the use of both models together. According to our systematic mapping review, there are 24 primary studies.
(2) DevOps shares several key aspects with CMMI, including: process improvement, measurement and metrics, continuous improvement, cross-functional collaboration, quality and compliance, data-driven decision-making, customer-centric approach, risk management, customer satisfaction, operational efficiency, software testing, and a broad systems approach.
(3) Several proposals have been presented by researchers for using both DevOps and CMMI to improve the software development process. Most of these proposals were frameworks or models for measuring DevOps maturity based on CMMI.
(4) It is possible to integrate DevOps with CMMI based on several factors, primarily the principle of integration, which is facilitated by the compatibility between them through their shared fundamental concepts. Additionally, there are models proposed by researchers for using both together, several of which have been successful in practice. Moreover, CMMI has expanded its process areas to align with the context of other models, including DevOps, due to the increasing desire of software companies to use both of them to improve the software development process.

**TABLE 7.** The comparison of models/instruments used DevOps and CMMI together.

| *Model/Instrument* | *Features/Focus Areas/Dimensions* | *DevOps Capabilities/ success Factors* | *Number of Dimension* | *Number of Levels* | *Levels* | *Demonstration* |
|---|---|---|---|---|---|---|
| Teixeira maturity model | People<br>Process<br>Technology<br>Culture | Continuous Deployment<br>Continuous Integration<br>Continuous Monitoring<br>Continuous Testing<br>Infrastructure as code<br>Feedback Loops | 4 | 5 | - | The applicability and usability of the artifact was demonstrated by means of an application test with three participant organizations. |
| Radstaak Maturity model | Automation<br>Collaboration & Communication<br>Culture<br>Measurement monitoring | Team structure<br>Process improvement<br>Feedback cycles<br>General automation<br>Environment provisioning (test, staging, production etc.)<br>Build automation<br>Deployment Automation<br>DevOps tooling<br>Documentation & configurations<br>Communication & Coordination<br>Collaboration<br>Reporting<br>Requirement management<br>Testing<br>Quality assurance<br>Monitoring<br>Code commit actions<br>Incident handling. | 5 | 5 | Initial<br>Managed<br>Defined<br>Measured<br>Optimized | Validated through expert opinions and put to practice in case studies. |
| Readiness model RMDevOps | RMDevOps is designed based on the investigated critical challenges CCH which plotting into levels. | The investigated best practices were mapping with every CCH as following:<br>Informal level:<br>- Reuse previous practices depending upon individual iterative.<br>Initial:<br>- Documentation and design code.<br>- Avoid resistance to adopt DevOps.<br>- Build knowledge.<br>Managed:<br>- Resolve issues about DevOps awareness.<br>- Flexibility issues due to rigid structure.<br>- Effective communication.<br>- Resource accountability.<br>- Disintermediation of roles.<br>Defined:<br>- Uncertainties such as integrated tools and practices.<br>- Heterogeneity in structure.<br>- Infrastructure maintainability.<br>- Use of immature automated tools.<br>- Threats modeling.<br>Quantitatively managed:<br>- Analyze legacy systems.<br>- Deep seated company culture.<br>- DevOps artifacts reusability.<br>Optimizing:<br>- Architecture refactoring.<br>- Balancing skills and high management strategies. | - | 6 | Informal<br>Initial<br>Managed<br>Defined<br>Qualitatively managed<br>Optimizing | A case study technique is used to validate the effectiveness of this model in real industry. |
| Samer I. Mohamed maturity model | Quality<br>Automation<br>Collaboration<br>Governance | Constructive communication environment<br>Processes focus on collaboration<br>Continuous deployment<br>Continuous testing (automated recoverability) | 4 | 5 | Initial<br>Managed<br>Define<br>Measured<br>Optimized | Not applied |
| IBM DevOps maturity model | People<br>Practices<br>Technologies<br>Information that support the software delivery process. | Release pipeline<br>Change & configuration Management<br>Dashboards/Analytics<br>Portfolio management<br>Requirements<br>Code<br>Test<br>Deployment<br>Provisioning | 4 | 4 | Practiced<br>Consistent<br>Reliable<br>Scaled | Validated by IBM |
| Capability Measurement Instrument for DevOps Teams | CALMS:<br>- Culture<br>- Automation<br>- Lean<br>- Measurement replaced by Monitoring<br>- Sharing | Intrapreneurship<br>Continuous improvement<br>Self-empowerment<br>Continuous software engineering<br>Infrastructure & configuration management<br>Artifact management<br>Architecture management<br>Security & access management<br>Lean process management<br>Change & operations management<br>Continuous planning<br>Customer-centric design<br>Requirement management<br>End-user monitoring & feedback<br>System monitoring & documentation<br>Knowledge sharing<br>Team collaboration | 6 | 5 | Automation, Lean, Monitoring (Definition adapted from CMMI capability levels): Incomplete – Performed - Managed Defined - Optimizing Culture, sharing (Definition adapted from Magdaleno CollabMM ): Initial - Ad-hoc - Planned Aware - Reflexive | The capability measurement instrument at hand was validated by applying a combination of domain expert evaluation and a field study |

(5) The integration of DevOps and CMMI leads to improvements in the software development process, thereby advancing the software industry.

## VII. THREATS TO VALIDITY

Similar to any secondary research procedure, it is nearly impossible to ensure that we have located every relevant

**TABLE 7.** *(Continued.)* **The comparison of models/instruments used DevOps and CMMI together.**

| | | | | | | |
|---|---|---|---|---|---|---|
| DevOps-RAF | DevOps-RAF main elements are the (organizational) Maturity Levels, Domains, Key Process Area (KDA), Key Performance Indicator (KPI), and their Capability Levels. DevOps-RAF domains are: Culture Agile development Agile operations Continuous integration | Project Documentation Outsourcing development Process Infrastructure Outsourcing operation Technical support Continuous integration/delivery Persons Collaboration and communication | 4 | 4 | Initial Intermediate Advanced Optimized | The proposed assessment framework was validated using the expert judgment, Later, a modified version of DevOps-RAF was pilot-tested on a development SME and an industrial customer. |
| DevOps requirement change management and implementation maturity model (DevOps-RCMIMM) | To develop the DevOps-RCMIMM levels, the critical barriers (CBs) were mapping to critical success factors (CSFs) through four RCM domains: - Knowledge Integration - Software Methodology - Human Resources Management - Technology Factors - Coordination - Project Administration | DevOps change management engineering Sharing information across distributed DevOps sites Requirement's elicitation and identification with DevOps teams Effort and cost estimation for change Clear change management strategy in DevOps Change management process awareness among DevOps team Governess and control of RCM activities Vision and goal for change Advance & uniform RCM infrastructure at DevOps GSD sites Financial stability in DevOps Change identification and validation in DevOps Frequent overseas site's response Change management automation in DevOps Effective RCM leadership in DevOps DevOps top management commitment DevOps RCM team motivation Accountability of RCM activities Determining the impact of demanded changes on DevOps Proper RCM task coupling in DevOps RCM process measurement across DevOps GSD sites Understanding of requested changes in DevOps Frequent communication and dialog in DevOps | 4 | 5 | Initial Managed Change management control Defined Optimizing | Not applied |
| Framework for DevOps process improvement | Practices Team Culture Measurement | Identify requirements of (tools – team – culture change) and quality measurement criteria. Apply DevOps tools and technology – prepare multi-function team – adopt culture changes – establish quality measurement criteria. Verify the application of DevOps tools – function the DevOps team - Establish collaboration culture and implement required changes – apply defined QAC with necessary the measurement process. Use the chosen DevOps tools and majority of processes are automated – well functioned team with knowledge skills, and responsibilities – make the culture shift with establishing perfect knowledge base. Continuous quality improvement with assessing progress against the goals and customer satisfaction. Release cycles are accelerated by automating all processes, and combining DevOps tools – strengthen the team with decision making and innovative skills - Further strengthen with collaboration with perfect communication – Fully Automated the quality measurement process with well-established QMC. | 4 | 5 | Initiate Beginning Intermediate Advanced Expert | Validated by industry practitioners |

document in the community. To reduce risks to validity, a number of steps were, however, performed insufficiently.

- Based on the recommenced guidelines for similar situations [15], a systematic mapping review process was employed to assure the inclusion of nearly all pertinent academic publications in the subject.
- By including a variety of synonyms for each phrase and articles that discuss combining CMMI and DevOps, we attempted to make our search string as broad as feasible. Not all of the relevant studies might be found, though, if the final search string does not include all of the synonyms that are currently in use. We mitigated this threat by reviewing the references of the final selected papers to include any additional pertinent works.
- A strict procedure for selecting studies was adhered, with defined inclusion and exclusion criteria, in order to reduce errors resulting from subjective evaluations. Nonetheless, the exclusion of articles written in languages other than English might have prevented the discovery of certain pertinent works.
- To increase the quantity of candidate publications, five reliable and well-known data sources (Scopus, IEEE Xplore, ACM Digital Library, ScienceDirect, and SpringLink) were selected.
- A precise data extraction template was established, and our findings were discussed in multiple brainstorming sessions in order to achieve data consistency and prevent bias in data extraction.

## VIII. FUTURE STUDIES

This systematic mapping review investigates the potential for integrating DevOps with CMMI by examining their commonalities and the successful models proposed for their combined use. We intend to conduct a future study addressing

the barriers and gaps between DevOps and CMMI to develop a model for their integration.

Future research should utilize empirical studies and real-world case analyses to tackle the identified gaps in the integration of DevOps and CMMI. Standardized frameworks must be developed to reconcile the rigid CMMI methodology with the flexibility of DevOps. Furthermore, exploring their role in this integration presents significant potential, especially given the rapid advancement of technologies such as artificial intelligence and cloud-native applications. Lastly, both academia and industry can benefit from defining and validating metrics to assess the effectiveness and return on investment of such integration efforts. This remains a promising area for future investigation.

## IX. CONCLUSION

A systematic mapping study of the integration between DevOps and CMMI suggests a large opportunity for synergistic improvements in software development processes. The combination of DevOps' emphasis on quick delivery and continuous improvement with CMMI's structured approach to process maturity provides a comprehensive framework for developing high-quality, efficient, and dependable software products. Starting from 355 papers retrieved from five well-known digital libraries, 24 primary papers were recognized as fulfilling the inclusion and exclusion criteria. Key findings show that, despite the increasing interest in both models, the compatibility between them, evidenced by common concepts, the potential for their integration, and the capability of this integration to improve the software development process, studies that address the use of both models together are still limited.

The data extracted from the selected 24 papers are considered a useful set of references for further analysis in any form of combining DevOps with CMMI. Besides, commonalities between them and proposed methods for using them together that were identified from this mapping study have provided many new ideas for research opportunities.

Finally, the convergence of DevOps with CMMI shows considerable promise for developing software development processes. By combining these capabilities, businesses can reach better levels of process maturity, operational efficiency, and innovation, resulting in superior software products and services. Further study and practical implementations and developing models are required to fully appreciate and standardize the benefits of this powerful integration.

## REFERENCES

[1] P. Buttles-Valdez, A. Svolou, and F. Valdez, "A holistic approach to process improvement using the people CMM and the CMMI-DEV: Technology, process, people, & culture, the holistic quadripartite," presented at the Softw. Eng. Process Group (SEPG) North Amer. Conf., Tampa, FL, USA, Mar. 2008.

[2] B. B. N. de França, H. Jeronimo, and G. H. Travassos, "Characterizing DevOps by hearing multiple voices," in *Proc. 30rd Brazilian Symp. Softw. Eng.*, Maringá, Brazil, Sep. 2016, pp. 53–62, doi: 10.1145/2973839.2973845.

[3] P. Narang and P. Mittal, "Hybrid model for software development: An integral comparison of DevOps automation tools," *Indonesian J. Electr. Eng. Comput. Sci.*, vol. 27, no. 1, p. 456, Jul. 2022, doi: 10.11591/ijeecs.v27.i1.pp456-465.

[4] L. E. Lwakatare, P. Kuvaja, and M. Oivo, "An exploratory study of DevOps: Extending the dimensions of DevOps with practices," M3S, Fac. Inf. Elect. Eng., Univ. Oulu, Oulu, Finland, Tech. Rep., 2016.

[5] T. Offerman, R. Blinde, C. J. Stettina, and J. Visser, "A study of adoption and effects of DevOps practices," 2022, *arXiv:2211.09390*.

[6] A. Ghantous and G. B. Gill, "DevOps: Concepts, practices, tools, benefits and challenges," in *Proc. Pacific Asia Conf. Inf. Syst. (PACIS)*, Langkawi, Malaysia, 2017, pp. 1–12.

[7] H. M. Thennakoon and D. A. Meedeniya, "Effort estimation strategies and practices of DevOps-based software development in sri Lanka," in *Proc. 11th Int. Conf. Inf. Commun. Manage. (ICICM)*, Tokyo, Japan, Aug. 2021, pp. 1–10.

[8] D. Teixeira, R. Pereira, and T. Henriques, "A maturity model for DevOps," *Int. J. Agile Syst. Manage.*, vol. 13, no. 4, pp. 464–511, 2020, doi: 10.1504/IJASM.2020.112343.

[9] R. Hernández, B. Moros, and J. Nicolás, "Requirements management in DevOps environments: A multivocal mapping study," *Requirements Eng.*, vol. 28, no. 3, pp. 317–346, Sep. 2023, doi: 10.1007/s00766-023-00396-w.

[10] I. Rubasinghe, D. Meedeniya, and I. Perera, "Traceability management with impact analysis in DevOps based software development," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Bangalore, India, Sep. 2018, pp. 1956–1962, doi: 10.1109/ICACCI.2018.8554399.

[11] M. A. McCarthy, L. M. Herger, S. M. Khan, and B. M. Belgodere, "Composable DevOps: Automated ontology based DevOps maturity analysis," in *Proc. IEEE Int. Conf. Services Comput.*, New York, NY, USA, Jun. 2015, pp. 600–607, doi: 10.1109/SCC.2015.87.

[12] J. F. Perez, W. Wang, and G. Casale, "Towards a DevOps approach for software quality engineering," in *Proc. Workshop Challenges Perform. Methods Softw. Develop.*, Austin, TX, USA, Jan. 2015, pp. 7–12, doi: 10.1145/2693561.2693564.

[13] M. Radstaak, "Developing a DevOps maturity model: A validated model to evaluate the maturity of DevOps in organizations," M.S. thesis, Fac. Eng. Technol., Univ. Twente, Enschede, The Netherlands, 2019.

[14] S. Milivojevic, "DevOps: A maturity model for small software companies," M.S. thesis, Fac. Organizational Sci., Univ. Belgrade, Serbia, 2021.

[15] C. J. Pardo, J. Guerrero, and E. S. Monsalve, "DevOps model in practice: Applying a novel reference model to support and encourage the adoption of DevOps in a software development company as a case study," *Periodicals Eng. Natural Sci. (PEN)*, vol. 10, no. 3, pp. 1–10, 2022.

[16] M. Zarour, N. Alhammad, M. Alenezi, and K. Alsarayrah, "A research on DevOps maturity models," *Int. J. Recent Technol. Eng.*, vol. 8, no. 3, pp. 4854–4862, Sep. 2019.

[17] L. Monteiro, D. S. M. P. Monteiro, W. Almeida, A. C. D. Lima, and I. S. Sette, "Methods of implementation, maturity models and definition of roles in DevOps frameworks: A systematic mapping," *IEEE Access*, vol. 8, pp. 1–10, 2020, doi: 10.1109/CSCI51800.2020.00327.

[18] J. Guerrero, C. Pardo, and C. Orozco, "DevOps ontology: An ontology to support the understanding of DevOps in the academy and the software industry," *Periodicals Eng. Natural Sci.*, vol. 11, no. 1, pp. 100–110, 2023.

[19] A. Kumar, M. Nadeem, and M. Shameem, "Assessment of DevOps maturity in software development organizations: A practitioners perspective," in *Proc. ICEASE*, 2022, pp. 438–443.

[20] Irigoyen and M. Montoni. (2020). *Capability Counts Series: Reaching CMMI Engineering Practices With DevOps*. [Online]. Available: https://cmmiinstitute.com/news/blog/capability-counts-series-reaching-cmmi-engineering

[21] CMMI Institute. (2019). *Taking DevOps to the Next Level With CMMI V2.0*. [Online]. Available: https://cmmiinstitute.com/resources

[22] J. M. S. Ruiz, F. J. D. Mayo, X. Oriol, J. F. Crespo, D. Benavides, and E. Teniente, "A benchmarking proposal for DevOps practices on open source software projects," 2023, *arXiv:2304.14790*.

[23] DORA Team at Google Cloud. (2021). *Accelerate State of DevOps*. Google Cloud. [Online]. Available: https://cloud.google.com/devops/state-of-devops

[24] E. Mueller. (2010). *What is DevOps?* The Agile Admin. [Online]. Available: http://goo.gl/mD3LU1

[25] B. Hornbeek, *Engineering DevOps*. Hoboken, NJ, USA: Wiley, 2019.

[26] S. Rafi, W. Yu, M. A. Akbar, S. Mahmood, A. Alsanad, and A. Gumaei, "Readiness model for DevOps implementation in software organizations," *J. Softw., Evol. Process*, vol. 33, no. 4, Apr. 2021, doi: 10.1002/smr.2323.

[27] (2010). *CMMI Product Team, CMMI for Development*. Version 1.3. Software Engineering Institute, Carnegie Mellon University. [Online]. Available: http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=9661

[28] ISACA. (2023). *CMMI Adoption Guidance*. [Online]. Available: https://www.isaca.org

[29] K. Doyle. (2018). *Introducing CMMI Development V2.0 to PAN-INDIA SPIN*. CMMI Institute. [Online]. Available: https://cmmiinstitute.com/resources

[30] C.-E. Orozco-Garcés, C.-J. Pardo-Calvache, and Y.-H. Salazar-Mondragón, "What is there about DevOps assessment? A systematic mapping," *Revista Facultad de Ingeniería*, vol. 31, no. 59, Mar. 2022, Art. no. e13896, doi: 10.19053/01211129.v31.n59.2022.13896.

[31] M. Gasparaite, K. Naudziunaite, and S. Ragaisis, "Systematic literature review of DevOps models," in *Quality of Information and Communications Technology*, vol. 1266. Berlin, Germany: Springer, 2020, pp. 184–198, doi: 10.1007/978-3-030-58793-2_15.

[32] S. Badshah, A. A. Khan, and B. Khan, "Towards process improvement in DevOps: A systematic literature review," in *Proc. 24th Int. Conf. Eval. Assessment Softw. Eng.*, 2020, pp. 427–433, doi: 10.1145/3383219.3383280.

[33] J. A. V. M. K. Jayakody and W. M. J. I. Wijayanayake, "Process improvement framework for DevOps adoption in software development," in *Proc. Int. Res. Conf. Smart Comput. Syst. Eng. (SCSE)*, Jun. 2023, pp. 1–7, doi: 10.1109/scse59836.2023.10214992.

[34] M. A. Akbar, A. A. Khan, S. Mahmood, and S. Rafi, "A vision of DevOps requirements change management standardization," in *Proc. IEEE 22nd Int. Conf. Softw. Qual., Rel., Secur. Companion (QRS-C)*, Dec. 2022, pp. 587–592.

[35] O. H. Plant, J. van Hillegerberg, and A. Aldea, "Design and validation of a capability measurement instrument for DevOps teams: A participatory action research approach," in *Agile Processes in Software Engineering and Extreme Programming* (Lecture Notes in Business Information Processing), V. Stray, K. J. Paasivaara, and M. Kruchten, Eds., Berlin, Germany: Springer, 2022, pp. 151–167, doi: 10.1007/978-3-031-08169-9_10.

[36] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Inf. Softw. Technol.*, vol. 64, pp. 1–18, Aug. 2015, doi: 10.1016/j.infsof.2015.03.007.

[37] F. M. A. Erich, C. Amrit, and M. Daneva, "A qualitative study of DevOps usage in practice," *J. Software: Evol. Process*, vol. 29, no. 6, p. e1885, Jun. 2017, doi: 10.1002/smr.1885.

[38] G. Rong, H. Zhang, and D. Shao, "CMMI guided process improvement for DevOps projects: An exploratory case study," in *Proc. IEEE/ACM Int. Conf. Softw. Syst. Processes (ICSSP)*, May 2016, pp. 1–10, doi: 10.1145/2904354.2904372.

[39] M. Gasparaite and S. Ragaişis, "Comparison of devops maturity models," in *Proc. CEUR Workshop*, Jan. 2019, pp. 65–69. [Online]. Available: https://ceur-ws.org/Vol-2470

[40] S. Mohamed, "DevOps shifting software engineering strategy-value-based perspective," in *Proc. Int. Conf. Softw. Eng. Appl. (ICSEA)*, 2015.

[41] A. Marnewick and J. Langerman, "DevOps and organizational performance: The fallacy of chasing maturity," in *Proc. Int. Conf. Inf. Syst. (ICIS)*, 2020.

[42] S. Ziaei, "DevOps metrics, objectives, and key results," *J. Softw. Eng. Res. Develop.*, vol. 10, no. 4, pp. 45–56, 2022.

[43] T. Neubrand and T. Haendler, "Development of a GQM-based technique for assessing DevOps maturity," in *Proc. 12th Int. Joint Conf. Knowl. Discovery, Knowl. Eng. Knowl. Manage.*, 2020, pp. 117–129.

[44] T. Mukange and J. Langerman, "The alignment between DevOps and business outcomes," in *Proc. Int. Conf. Eng. Emerg. Technol. (ICEET)*, Oct. 2022, pp. 1–6.

[45] L. Marrero and H. Astudillo, "DevOps-RAF: An assessment framework to measure DevOps readiness in software organizations," *IEEE Access*, vol. 9, pp. 104567–104579, 2021.

[46] K. Port, B. Taber, and P. Emkani, "Investigating effectiveness and compliance to DevOps policies and practices for managing productivity and quality variability," *IEEE Trans. Softw. Eng.*, vol. 50, no. 3, pp. 1–15, 2024.

[47] M. Z. Toh, S. Sahibuddin, and M. N. Mahrin, "Adoption issues in DevOps from the perspective of continuous delivery pipeline," in *Proc. Int. Conf. Softw. Eng. Adv. (ICSEA)*, 2019, pp. 123–131.

[48] C. Orłowski, T. Deręgowski, M. Kurzawski, and A. Ziółkowski, "Building dedicated project management process basing on historical experience," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 27, no. 4, pp. 623–637, 2017.

[49] T. Engdashet, R. J. Machado, and D. Midekso, "Integrated framework of agile and CMMI: An alternative," in *Proc. IEEE Int. Conf. Agile Softw. Develop. (ICASD)*, Feb. 2016, pp. 89–97.

[50] B. Phifer, "Next-generation process integration: CMMI and ITIL do DevOps," *Cutter IT J.*, vol. 24, no. 6, pp. 10–15, 2011.

[51] P. Bahrs, "Adopting the IBM DevOps approach for continuous software delivery," *IBM Syst. J.*, vol. 52, no. 1, pp. 89–102, 2013.

[52] M. Große-Rhode, "On model integration and integration modeling—Introduction to the subject area integration modeling," in *Integration of Software Specification Techniques for Applications in Engineering*. Berlin, Germany: Springer, 2004, pp. 567–581.

[53] F. Selleri Silva, F. S. F. Soares, A. L. Peres, I. M. D. Azevedo, A. P. L. F. Vasconcelos, F. K. Kamei, and S. R. D. L. Meira, "Using CMMI together with agile software development: A systematic review," *Inf. Softw. Technol.*, vol. 58, pp. 20–43, Feb. 2015, doi: 10.1016/j.infsof.2014.09.012.

[54] P. McMahon, *Integrating CMMI® and Agile Development: Case Studies and Proven Techniques for Faster Performance Improvement* (The SEI Series in Software Engineering). London, U.K.: Pearson, 2010.

[55] P. Keil and M. Fritzsche, "Agile methods and CMMI: Compatibility or conflict?" *e-Informatica Softw. Eng. J.*, vol. 1, no. 1, pp. 7–18, 2007.

[56] J. Glazer, "CMMI® Or agile: Why not embrace both?" Softw. Eng. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep., 2008.

[57] KPMG International. (2018). *A Capability Improvement Framework to Disrupt, Transform, and Innovate*. KPMG Insights. [Online]. Available: https://home.kpmg.com/in

[58] V. Henríquez, A. M. Moreno, J. A. Calvo-Manzano, and T. S. Feliú, "Agile-CMMI alignment: CMMI V2.0 contributions and to-dos for organizations," *IEEE Comput.*, vol. 53, no. 6, pp. 45–52, Jun. 2020.

[59] Research and Assessment (DORA) Team at Google Cloud. (2024). *Accelerate State of DevOps*. [Online]. Available: https://cloud.google.com/resources/devops/state-of-devops



**RANIA RAJAB** received the B.S. degree in informatics engineering from Tishreen University, Syria, in 2003, and the M.S. degree in technology management from Syrian Virtual University (SVU), Syria, in 2018. Currently, she is pursuing the Ph.D. degree in technology management. Since 2020, she has been an Assistant Teacher for a number of software engineering courses with Syrian Private University (SPU). Her research interests include software engineering, software process improvement, CMMI, and DevOps.



**MOUHIB ALNOUKARI** received the B.S. degree in informatics engineering from the Higher Institute for Applied Sciences and Technology, Syria, in 1989, the D.E.A. degree in informatics from Montpellier University, France, in 1993, and the Ph.D. degree in software engineering from Damascus University, Syria. He is currently an Assistant Professor with the Department of Software Engineering and Information Systems, Faculty of Engineering, Syrian Private University (SPU), Syria. He co-authored many academic books in computer science and more than 20 articles. His research interests include software engineering, business intelligence, data mining, data warehousing, and agile methodologies.