

DEVOPS ADOPTION IN INFORMATION SYSTEMS PROJECTS; A SYSTEMATIC LITERATURE REVIEW

J. A. V. M. K. Jayakody^{1,2} and W. M. J. I. Wijayanayake³

¹Faculty of Applied Sciences, Uva Wellassa University, Sri Lanka,

²Faculty of Graduate Studies, University of Kelaniya, Sri Lanka,

³Faculty of Science, University of Kelaniya, Sri Lanka

ABSTRACT

The word DevOps derives from two different words Development and Operations. DevOps has recorded as an interesting and novel approach adopted to the commonly used Agile software development methodology. It raised agility of the software development process. Practical issues of Agile methodology emphasize the requirement for collaboration of software development and operating teams. This collaboration completed by the DevOps approach engages with the Agile methodology to improve the quality, performance, and speed of the software developments. Since DevOps is an accentuating approach in the software development industry, this research aimed to conduct a literature review to study the evolution of the DevOps approach and its adoption in information systems projects. This target has accomplished by reviewing the Agile methodology, issues of the Agile methodology, DevOps approach, challenges and overcoming strategies of DevOps, and success factors of the DevOps approach. Finally, the paper provides better acquaintance about the DevOps adoption in Information System projects developments.

KEYWORDS

Agile, DevOps, DevOps Challenges, DevOps Overcoming Strategies, DevOps Success Factors

1. INTRODUCTION

An information system is a collection of components for collecting data, processing information, storing information, and making information available when it is needed. It involves Information technology, management, and organizational dimensions, which are critical for converting human organization into virtual. The success of information systems mainly depends on the success of the information systems development process [1]. Diverse software development methodologies are practiced in the software development industry since each methodology provides challenges for the success of information systems. Traditional software development methodologies were applied sequential processes over the software development life cycle for developing the software. It consumes more time to deliver the final product to the customer, and customers could not be able to change their requirements while the development because of the inflexibility of the development process. Such complications derived software designers from engaging a new software development methodology called "Agile."

Agile software development methodology divides the development process into many sprints and parallelly develops each sprint by reducing the problems raised by traditional software development methodologies. It speeds up the development process while improving quality of the software by building a good relationship between software developers and customers. However, the benefits provided by Agile methodology make it as most popular software development methodology in the industry. While practicing the Agile methodology, software designers could

understand some remaining problems, and it pushed them to introduce new approaches to the Agile methodology. As a result of this situation, the DevOps approach introduced into the software development industry mainly focuses on maintaining a good relationship between software development and operation teams. This vital relationship confirms the faster development, quality assurance, and easy maintenance of the information systems, dropping the disadvantages and challenges created by the Agile software development methodology.

This literature review study aims to survey the evolution of the DevOps approach and its adoption in information systems projects. The aim was achieved by completing two objectives. Study the Agile software development methodology and its associated problems, which caused the introduction of the DevOps approach. Furthermore, discuss the adoption of DevOps approach in information systems projects by observing the challenges, overcoming strategies, and critical success factors.

The research objectives have accomplished by answering the following research questions.

- Q1 - What is Agile Software Development Methodology?
- Q2 - What are the issues of Agile Software Development Methodology?
- Q3 - What is the DevOps approach?
- Q4 - What are the challenges and overcoming strategies of the DevOps approach?
- Q5 - What are the critical success factors of the DevOps approach?

The remainder of this paper consists of the methodology applied for conducting the research and the survey results. Moreover, it concludes with a discussion about the study and recommendations for future researchers.

2. RESEARCH METHODOLOGY

The research followed a systematic literature review method, which helps to survey the state of the art of the research areas that are not yet mature. Relevant publications downloaded from the Google Scholar, Emerald Inside, Web of Science, and Google Search Engine using the search keywords formed by research questions as “Agile”, “DevOps”, “Agile” AND “Issues”, “DevOps” AND “Significance”, “DevOps” AND “Challenges”, “DevOps” AND “Overcoming Strategies”, “DevOps” AND “Critical success factors”. According to that, 312 papers have downloaded and the following inclusion and exclusion criteria were applied to select the most relevant 218 papers for fulfilling the research purpose.

Inclusion Criteria

- Literature discusses the Agile Software Development Methodology
- Literature discusses the issues of Agile Software Development Methodology
- Literature discusses the evolution of DevOps
- Literature discusses the challenges of DevOps adoption
- Literature discusses the overcoming strategies of DevOps challenges
- Literature discusses the significances of DevOps adoption
- Literature discusses the critical success factors of DevOps adoption

Exclusion Criteria

- Literature not related to the purpose of the study
- Inaccessible literature

- Duplicated literature

As the next step, more related 196 publications were identified to fulfill the research objectives with reading paper titles and keywords. Afterward, the selected papers' abstracts helped screen the most relevant publications from the selected papers. Finally, the study was conducted by reading the entire paper of the most relevant 61 papers, which selected from this systematic approach. It helped to fulfill the aim of the research by answering the research questions. As the last, research provides future recommendations for the researchers who are interested in this area.

3. RESULTS AND DISCUSSION

3.1. Agile Software Development Methodology

"Able to move quickly and easily" is the literal meaning of Agile, and relating to software development, Agile divides tasks into short work phases and releases the plans frequently. Software development companies have moved into teamwork since it reduces completion time and increases software quality. A team can define as "a group of individuals working together who are dependent upon one another and have one or more tasks to perform collectively that result in a specific outcome" [2]. The agile software development methodology is well suits for group work and it delivers high-quality products within a minimum period. This leads software companies to earn their customer's satisfaction [2]. Agile makes decision-making power decentralized with team members [2] and customers continuously involved in this decision-making process. As well as, the Agile methodology allows software development teams to make effective communication, analyze risk, reduce cost and reduce bugs, which helps to increase the quality with the faster delivery [3].

As highlighted by Ahimbisiwe et al. (2015), selecting software development methodology involves two main categories called traditional plan-based approach and open agile approach. Traditional plan-based approaches base on plans, but the open agile approach is less planned and suites dynamic environments that change quickly [4]. Unpredictability is the nature of the world. This challenge can address by the Agile method by recognizing the value of competent people and their relationships over the software development lifecycle [5]. Agile methodology does not follow a sequential process for developing software like traditional software development methodologies. Because of this sequential process of the traditional software development methodologies, the end product is not available to customers until the very end of the project. It could take months or years to complete the project [2].

Moreover, it was not allowed rapid changes and was poorly supported to increase the efficiency of the software development process. According to previous studies, user involvement is essential over the software development life cycle and the origin of agile methodology caused those requirements. Agile methodology reduces these issues by dividing the software development process into the regular short cycle of deliverables called sprints [6]. These parts are more manageable than the managing software as one deliverable. Some industries called these sprints "user stories" [7]. The output of each sprint can make available to customers. They can suggest changes and recommendations while developing the software [6], and this helps to improve customer satisfaction by adopting their requirements changes [2]. Therefore, as Abrahamsson P. and et al. (2002) discussed, the Agile methodology is suitable for the environment when small software parts release in rapid cycles, customers and developers work together, and able to make last moment changes [8]. And also, agile teams are small teams consisting of less than nine members and these teams work iteratively to develop software sprints in a short period, as short as two weeks. First, they set objectives at the start of each sprint and work for short time

objectives with short-term decisions and slight documentation [2]. Drury and Meghann (2014) argued that this helps software companies deliver quality products within a minimum time period. As Dings, Dybo, and Moe (2010) discussed, agile methodology has become an umbrella term for the changes in planning and coordinating works by software developers, their communication strategies, and how they organize as teams [5].

When comparing the traditional software development methodologies, agile methodology benefits many areas, such as; customer collaboration, work processes for handling defects, learning in pair programming, thinking ahead for management, and focusing on current work for engineers [5]. Agile methodology allows software developers to manage their development process without preparing for future features in the current iteration since they follow an iterative and adaptive life cycle that positively impacts quality [4]. Ibrahim (2007) has grouped these benefits into twelve areas and published them as a bunch of agile principles. He explained that agile could make customer satisfaction through early and frequent delivery. Delivery cycles are short, and working software can measure project progress. As same as business people and developers working together and projects built around motivated people. Self-organizing teams, continuous attention, and face-to-face communication are also special features of the agile methodology [9]. Similarly, a study conducted to observe the usability engineering and agile methodology proved that software usability well fits with the agile software development methodology [10].

Software development companies follow many types of agile software development methods. According to Rasnacic A. & Berzisa S. (2016), it consists of more than twenty different methods, and suitable methods can select considering the software type, organization, and its employees [3]. And they have discussed the most popular methods like Scrum, Extreme programming, Kanban, Lean software development, Feature-driven development, Agile unified process, and Dynamic systems development method. More than the above methods, Abrahamsson, P., Outi S., Jussi R., and Juhani W. have discussed the Crystal family of method and the Adaptive software development method. Not only that, they have presented the open-source development method as the most popular agile software development method, and agile modeling and pragmatic programming method identified as non-popular methods, but the agile community has taken considerable attention. [8].

3.2. Issues of Agile Software Development Methodology

Agile software development methodology could provide more benefits to the software development industry, and it solved many issues raised with the traditional software development methodologies. However, existing research studies have reported more failures in the successful implementation of the software projects which followed agile software development methodology. This forced researching the issues of agile software development methodology and many challenges reported in the existing studies. Since the agile software development methodology solved issues of the traditional software development methodologies by mainly focusing on teamwork, it derives many benefits. However, it became one of the reasons for the reported issues. Fitriani et al. (2016) have explained team management as the first and most significant challenge out of thirty challenges of agile software development methodology [11]. The agile methodology depends on the collective ability of the team to solve problems, and all team members should be skillful and valuable stakeholders [2]. Reasons for the issues of teamwork observed as the problems with allocating team members, coordinating them for the team activities, disagreements within the team, changing their role [11], team development, and maintenance tasks in teams [12]. Not only that, resistance to change is another problem for group work raised by the reasons, the team consists of individuals who have different behaviors, cultural differences of team members from different nationalities [13], and their geographical

differences [11]. Rasnaxis & Berzisa (2016) also explained that conflicts and other problems of the team directly affect quality of the software [3]. Agile methodology provides limited support for software development projects which consist of large groups [9]. According to Hoda and Murugesan (2016), achieving cross-functionality has been identified as a team-level challenge of agile methodology since the inability of team members to work across various technologies and functional areas [11]. They have also explained effective estimation as another team-level challenge. Because of the team works, it is difficult to estimate the performance of managers or leaders who are not included in the team [14]. L. F. Amorim and et al. (2020) have discussed the importance of improving the effective communication, motivation of team members, collaboration among members for reducing unhappiness of the Agile teams [67].

According to Fitriani et al. (2016), practicing individuals to work as a team, finding good leaders for all the teams, and motivating all team members are complex. It creates challenges for the software designers who follow the agile methodology [11]. Agile team members can face peer pressure and too much stress since agile methodology forces them to practice standup meetings, onsite customer meetings, retrospectives, pair programming, and frequent informal meetings. They must have good communication, social and presentation skills [15], [13]. This can lead to a high member turnover [11]. These agile requirements lead to forming an open working environment in the software development company. However, it causes disturbances in concentrating on software development [16]. Software development team members used headphones to solve this problem, and again they feel that it is challenging to wear headsets over a long time [13].

As stated by the previous study, software usability well fits with the agile software development methodology. But they have argued that this advantage can be obtained only by including usability specialists in all the teams [10]. Similarly, software developers need to be "masters of all trades" [13]. In other words, the lack of business knowledge of the developers [13], [15] and the lack of information systems knowledge of the customers [9] leads to software project failures that follow agile methodology. Because, agile methodology is based on high interaction between customers and developers [15]. The usability and handling of other non-functional requirements are also problematic since customers are unfamiliar with the software development field [9]. This also proves by arguing that software developers must have a broad knowledge of all aspects of software development and specialized knowledge of their specialized area to run their groups into success [15].

Customer involvement throughout the software development period is one of the main features which generates benefits of agile software development methodology. This allows the customer to change their requirements, and some critical requirements can delay gathering [14]. Similarly, it is not easy to analyze and prioritize gathered requirements according to their significance [11], and customer opinion can be changed based on the person [9]. However, these ambiguous requirements lead to poor software quality and overrun the schedules [13]. Similarly, researchers observed that software designers face challenges in collecting ideas from the customers when needed since customers are not available at any time [16].

Most researchers have explained the benefits of agile methodology raised by poor documentation. Nevertheless, some studies discussed the challenges created by this poor documentation. According to Ibrahim (2007), the agile methodology does not provide required documents. This leads to wrong planning and technical decisions, and some decisions are not tracked and documented [17]. So no documentary evidence [11]. Not only that, poor documentation is challenging for new software development team members and members who work on the part of the project [16].

Similarly, task dependency reduces project performance and delays product delivery [11]. Not only that, it can help to cancel some project sprints [14]. High customer involvement creates a challenge for software designers as complex to reuse the software [9]. Senior management support is significant for achieving the success of agile projects. However, agile software developers face challenges obtaining their support [11], which has presented as a project-level challenge [14].

Studies on the challenges of agile software development methodology proved that issues remain in the software development industry while using this agile methodology. Most of the problems could solve by improving the collaboration between software developers and software operators[18].

3.3. Evolution of DevOps Approach

Agile software development methodology converted software development practices into a new era. It could solve the problems of traditional software development methodologies. However, software designers are recognized unsolved issues of the software development process while practicing this agile methodology. It forced software development practitioners to increase the collaboration between software developers and software operators by improving agile software development methodology with a new approach called DevOps (Development and IT Operations). Hemon et al. (2020) explained that software developed based on the agile methodology ensures the meeting of market needs. However, it could identify a bottleneck in the operations function that coordinates the actual release of software into the market. Always this operations function not align with the development function. This problem again increased the software delivery time to the customers. Nevertheless, the primary focus of the evolution of software was the speed of delivery of the software projects [66] . It highlights the need for a close relationship between the developers and operators. Banica et al. (2017) explained that the separation of software development and software operation leads to failures in meeting the project deadlines since it delays discovering the errors [19]. DevOps creates this continuous bond of the development and operations functions [20], [18].

The software development industry has recorded the DevOps concept since 2009 [19]. Different opinions are available, and no standard definition available for this DevOps concept. Some researchers have introduced DevOps as a conceptual framework, and others have presented DevOps as a specific skill set. Some of them explained that DevOps is the third generation of software development methodology which continues from the second generation, agile software development methodology [19]. In other words, DevOps is considered as an extension of agile methodology [7].

Dyck, Penners, and Licher (2015) defined DevOps as "an organizational approach that stresses empathy and cross-functional collaboration within and between the development and operation teams in software development organizations" [21]. Furthermore, according to Maximilien De Bayse (2018), DevOps is a software development method that extends the agile philosophy to rapidly produce software products and services and to improve operations performance and quality assurance [22]. Moreover, DevOps has been defined as "a software development methodology which looks to integrate all the software development functions, from development to operations, within the same cycle" [19]. Not only the technical definitions but also DevOps is defined as "an information technology mindset that encourages communication, collaboration, integration and automation among software developers and IT operations to improve the speed and quality of delivering software" [19]. However, Banica et al. (2017) research to study whether DevOps is a software development methodology stated that DevOps is not a software development methodology since no standard definition is available and no main characteristics

are defined. It uses agile principles by adding further operations and functionality for the deliverables.

DevOps overcome the gap between market needs and traditional software development methodologies [23] by breaking down the walls between traditional development methodologies versus operations silos [24] with the help of enhanced collaboration and communication. While most researchers defined DevOps as development and operation teams working together, Debois (2011) asks to think systematically. According to him, DevOps must apply to the entire organization, not only to the development and operation teams [25]. According to that, Devops is "about aligning the incentives of everybody involved in delivering software, with a particular emphasis on developers, testers, and operations personnel" [25].

Anderson (2019) provided another opinion on DevOps: "DevOps is the embodiment of technical and social skillsets necessary to merge development and operations functions, such as automated development and deployment, and system monitoring needed to achieve continuous delivery." Not only that, Riungu-Kalliosaari et al. (2016) defined DevOps as "a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal software development while ensuring high quality" [24].

The agile software development team consists of designers, testers, and developers [7], and they are always in contact with the customers [27]. DevOps pull agile methodology one step further by introducing software operators who support the transition between software and implementation into the software development team [7]. Furthermore, Mohammad (2017) argued that DevOps is a composition of instruments, cultures, and practices which divides silos into teams.

According to Banica et al. (2017), DevOps includes key primary directions such as collaborative culture, automation, service, and quality assurance. They explained about three phases followed by DevOps teams for developing software. First, they identify the clients' requirements, then build software components by the development team and test them by the operation team as the second step. This stage must base on a well-defined project plan, including incremental iterations/sprints. Each sprint should consist of the purpose, timeline, responsible members, and deliverables. The team continues to the next stage after confirming that there are no errors or incomplete components. The last step is to deliver developed components to the customer and take their feedback [19]. Meanwhile, research conducted for studying the relationship between DevOps and software quality discussed DevOps with eight features; Culture, Sharing, Fast feedback, Automation, Continuous delivery and deployment, Measurement, Software architecture, and DevOps practices. Thus, software designers need to pay attention to these eight features to make the success of DevOps [28].

However, DevOps affect and change the organizational culture by improving the interaction between software developers and operators [28]. Samarawickrama and Perera (2017) presented a framework that shows the application of the DevOps approach in the software design domain [29]. As shown in Figure 1., they explained that DevOps integrates the development domain and operation domain. Developers can reduce the issues of software applications by considering all the issues reported in the software testing stage, maintenance stage, the help desk stage, and other stages [29]. Correspondingly, a literature study conducted for the DevOps approach has summarized about eight features of DevOps as; communication and collaboration, continuous deployment, continuous delivery, automated pipeline, continuous planning, continuous feedback, rollback code, and quality assurance [30].

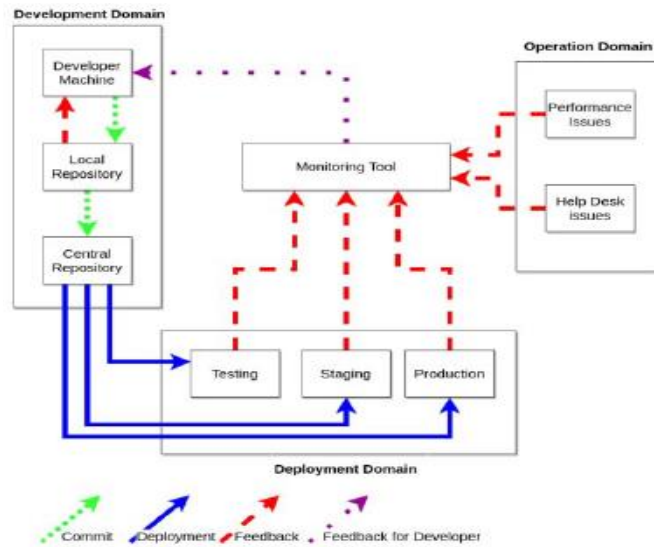


Figure 1. Framework for DevOps

Mainly, few researchers have studied DevOps adoption in the Sri Lankan software development industry. When considering the DevOps approach and the Sri Lankan software development industry, there are four crucial factors to consider with implementing DevOps: culture, automation, measurement, and sharing [31].

3.4. Significance of the DevOps approach

An in-depth case study conducted in an organization with experience in DevOps argues that DevOps leads to great smartness for the Information Systems through the soft skills and pattern of collaboration of the software teams [20]. Similarly, DevOps increase the openness of all teams and promote innovations by improving the communication between software developers and operators [7]. Similarly, DevOps' real-time monitoring and team cooperation practices help to assure the software quality by detecting real-time problems of customers [19]. DevOps well equip companies with knowledge of customer preferences, and they can meet the market demand and capture new opportunities [24]. Furthermore, Riungu-Kalliosaari et al. (2016) argued that it helps companies analyze different ideas quickly and make precise decisions. Jayakody & Wijayanayake (2021) explained that it mainly reduces the project completion time, improves software quality, and improves customer satisfaction [32].

According to Mishra and Otaiwi (2020), the main focus of DevOps is to increase the development speed, frequency, and quality. Mainly they have argued that DevOps can reduce the pressure on software designers to maintain the software quality. Furthermore, they stated the importance of researching the DevOps influence has on the software quality factors such as efficiency, usability, maintainability, and portability [28], [24]. While improving the quality and performance of software applications, DevOps can enhance end-user satisfaction and support simultaneously deploying software across different platforms [29]. Same as the quality, DevOps provides many advantages under its main factor listed as continuous delivery. It enables software development companies to take faster feedback from the customers [24], get better visibility into the delivery process, deliver software more effectively, reduce the cost of deployment, and mainly reduce the risk for software failures [25]. Not only that, DevOps can improve the reliability, stability, resilience, and security of the software [23]. And also, DevOps provides advantages to the software development companies by improving the ability to add more features

to the final software [24]. Again, it helps to increase customer satisfaction. Similarly, teams with diverse capabilities use their broader set of capabilities to improve the productivity of the outputs [24].

When considering the Sri Lankan software development industry, software professionals have stated that DevOps can positively impact software quality, responsiveness to business needs, and agility to new technologies [31]. These various advantages provided by the DevOps approach have increased the demand, and many software development companies are adopting the DevOps culture along with the agile methodology [33].

3.5. Challenges and Overcoming Strategies of the DevOps Approach

Previous studies have verified lots of benefits offered by the DevOps approach. However, DevOps practitioners have again faced some issues with adopting this new concept along with the agile software development methodology. Few researchers focused on this industry experience.

Many researchers have discussed software quality improvement as an advantage provided by the DevOps approach. However, recent empirical studies demonstrate that it is challenging to balance quality and software development speed [34], [35]. DevOps speeds up the development process while reducing the software development time, and this is challenging to improve and maintain the software quality. According to Fazal-Baqaie et al. (2017), it is challenging to achieve transparency on quality delivered by different teams [36]. Not only that, guidelines on developing high-quality logging code are not commonly available [37].

Meanwhile, in developing software based on the agile methodology and the DevOps approach, a challenge was raised with the problems of testing practices since DevOps changes the whole software testing process [38]. No perfect testing tools are available in the industry [39] since DevOps is a novel approach to the industry. As same as, challenging to find expertise to execute these processes [40], and the testing process takes more time [41]. Because of these problems, software developers face challenges in developing high secured software. Most of the related studies reported that providing good security to the software is the main challenge created by the DevOps approach [39], [42], [43].

Because of the lack of knowledge and experience in the DevOps approach, it is challenging to find DevOps experts in the software development industry [32]. Moreover, available tools and technologies for DevOps are very complex and difficult to use [44]. These problems force recent researchers to identify a DevOps challenge as lacking technical infrastructure [45],[39]. Lack of awareness among the software design team members about this new approach creates diverse challenges [46], [47]. The lack of top management support for this newest approach is a massive challenge for many researchers [46], [42], [35].

Besides the technical problems, DevOps practitioners could identify psychological problems while improving the collaboration and communication between software development and operation teams. Previously these teams were separated, and they operated in different locations. This situation provides challenges to DevOps teams for improving communication. Furthermore, most researchers focused on the resistance to change as a primary challenge for the DevOps approach. Software development and operation teams have to change their habits, behaviors, and thinking patterns. Nevertheless, it is not easy to change the behavior and habits of people [48]. Similarly, it is challenging to change the culture and social factors of the organizations to adopt the DevOps approach [46],[49], [50]. Not only the social and cultural changes, but DevOps also change the organizational process for improving team collaboration. However, software designers face

challenges while changing the organizational processes [51], [52]. Similarly, achieving the completability between legacy systems and the DevOps approach raises challenges for software development companies [51]. Difficulties for reproducing the complex technology environments needed for DevOps [48], [39] and using this complex technology [50], [40] are also identified as the challenges for DevOps adoption.

A limited number of researchers focused on similar studies for the Sri Lankan Software industry. According to them, while DevOps is considering to reduce the completion time and increase the software quality, it can involve hidden costs and generates problems related to the budget [53]. Perera, Bandara, and Indika (2017) stated that the Sri Lankan software industry also confirmed the challenge of finding DevOps experts. Changing the mind of top-level managers and employees takes some time [53]. Applying employees to teamwork, knowledge sharing, and ego in sharing knowledge are also identified as the DevOps adoption challenges in the Sri Lankan software development industry [53]. When it comes to another research conducted by analyzing opinion of the Sri Lankan DevOps practitioners, changing deep-seated company culture to support DevOps adoption and achieving a secure DevOps development process are the most critical challenges. Also, they have argued that the DevOps approach makes the software design process complex. [32].

Since few studies focus on this area, similar studies have provided a minimal number of suggestions for the above challenges. Establishing communication, platform, procedures, and tools for enhancing communication between software development and operation teams can apply for reducing the challenges of DevOps adoption [54],[55],[56]. Furthermore, based on the ideas of DevOps practitioners, this is the primary overcoming strategy that can apply to the Sri Lankan software industry [32]. Not only that, problems of the DevOps adoption solve by rearranging the software development groups with including DevOps expertise and improving knowledge about the DevOps approach of the existing team members [57], [55], [54]. Similar to the Sri Lankan software development context, DevOps adoption challenges can mitigate by applying a framework called CALMS. The framework led software developers to focus on Culture, Automation, Lean, Measurement, and Sharing of the software and software development teams. Similarly, communicating and celebrating success of the DevOps process assist and motivate software design team members to engage the DevOps practices. [32], [53].

3.6. Critical Success Factors of Adopting DevOps in Information Systems

DevOps is a new approach to the agile software development methodology, and it takes agile methodology into a new step. However, since this DevOps approach is an emerging trend in the software development industry, no more researchers have studied the essential factors that help reduce the software failures created by this DevOps approach. Very few studies have discussed those factors and presented them as the critical success factors for DevOps adoption. However, the critical success factors must have verified significant impact on the success of the continuous practice, verified significant improvement, and success and performance should be measured [58]. DevOps approach, along with the agile methodology is mainly based on teamwork. These teams must be cross-functional teams with members from development and operations teams. Every team member must have a good idea about DevOps approach adoption [59], [60] and their responsibility within the teamwork. Dumoulin (2017) argued that this is the fundamental success factor for DevOps.

According to the literature review study, a core factor for the success of DevOps is collaborative culture [51]. In other words, according to Blueprint (2017), connecting the silo culture is the first success factor of DevOps. A collaborative culture enables by the core categories as automation, transparency and sharing, continuous measurement, and quality assurance [51], [59]. Again, these factors have confirmed by the Blueprint (2017). Emphasizing the culture is more critical to

DevOps success than the tools [60]. Not only that, it has presented five critical success factors for DevOps adoption as; collaborative culture, closing the loop between monitoring and planning, measuring success based on the entire technology ecosystem, having practical DevOps goals, and leveraging technology to automate and orchestrate [61]. As same as, Stray et al. (2019) identified and published DevOps principles as knowledge sharing, automation of development and testing [60], shared responsibility, and continuous software development [62]. Similarly, communication between the team members, specialized team members, and customer involvement should be considered as critical factors for the success of DevOps [18]. Similarly, real-time feedback from the related stakeholders and a good security pipeline also provides benefits to the DevOps approach [60]. Not only that, according to Akbar et al. (2020), measuring the progress before planning the next improvement is also important for reducing the problems of continuing the development process. It is better not only to focus "now" but also to look at the maintenance services after the system delivers to the customer [63]. Based on the survey conducted to find the critical success factors of DevOps, there are no verified critical success factors available. It has grouped and presented five factors as critical for the DevOps approach as per resistance to change, quality, customer involvement, text complexity and source code control, and pace [64].

More than the above factors, similar studies reported about the critical skills essential for the success of DevOps adoption. Establishing technically and non-technically skilled DevOps teams is significant for reducing failures [60]. Analytical skills are required for monitoring and analyzing the problems and for better management. Functional skills help to understand the business needs and fulfill the final goals. The team's success mainly depends on the correct and quick decisions. It is forced by decision-making ability composed of self-confidence and responsible behaviors. Not only that, but advisory skills are also crucial for understanding and complying with agreements. It is also vital to have good knowledge and experience in testing codes with the DevOps team members to prove the correctness of the codes [65].

Non-technical skills are also significant for achieving the success of DevOps software development projects. Hemon et al. (2020) said that sometimes non-technical skills are more important than technical skills since DevOps is mainly based on teamwork and always development and operation teams attached to their work. This DevOps approach applies to the Agile software development methodology and also runs on the continuous communication between the customers and software developers. As non-technical skills, communication skill is essential for sharing information with the team members [20], [25]. No use of other skills not enabled by the social skills to share the knowledge and communication within the team [65]. Team members must be motivated in their work, and they must be adaptable to different situations and different working environments [20]. They must be flexible and responsible [20], [25] for their duties. Interpersonal skills, the ability to organize, and teamwork skills help build strong team member relationships to achieve the team objectives [25]. These all sound good. However, the success of DevOps adoption is not realized without creativity skills [20] and having a good leader for the DevOps groups [25], [60].

4. CONCLUSION

The study aims to survey the evolution of DevOps approach and its adoption in information systems projects and the aim has achieved by conducting a literature review study. DevOps is a spotlighted word in the current software development industry and it derives from two words as Development and Operations. DevOps emerges in the software development industry as a new approach to the Agile software development methodology. The agile methodology could comprise agility into the software development process by reducing the gap between traditional software development methodologies and the information systems market requirements. Mainly,

it runs on the team works and parallel development process, reducing the development time and minimizing the final product delivery time to the customers. Similarly, agile methodology generates a good relationship between the software developers and customers. Therefore, customers can change their requirements while developing the system and this forces to improve the software quality. It could also observe various advantages provided by Agile methodology, and those benefits make known Agile methodology as the most commonly used methodology in the software development industry. While practicing this Agile methodology, software developers could face issues that highlight the importance of improving the communication between software development and operating teams. This communication mainly aids to increase the software quality by improving efficiency, usability, maintainability, and portability.

DevOps can satisfy the customers by improving the reliability and the performance of information systems. Likewise, there are lots of DevOps benefits have discussed in similar studies. However, as with the benefits, few researchers have observed the challenges faced with adopting the DevOps approach. The primary and most commonly stated challenge is changing the deep-seated culture of the software development teams and their companies. Still, the researchers have suggested very few overcoming strategies for facing those challenges. This study discussed the DevOps approach with its challenges, overcoming strategies, and critical success factors, which help reduce the failures of information systems projects. Considering the highlights of this survey, it recommends future researchers in developing a model that helps to better application of the DevOps approach that aligns with the Agile software development methodology for attaining the success of information systems projects.

REFERENCES

- [1] J. Varajao, 'The many facets of information systems (+ projects) success', *Int. J. Inf. Syst. Proj. Manag.*, vol. 6, no. 4, pp. 5–13, 2018, doi: 10.12821/ijispm060401.
- [2] G. Drury and L. Meghann, 'Performance on agile teams: Relating iteration objectives and critical decisions to project management success factors', *Inf. Softw. Technol.*, vol. 56, no. 5, pp. 506–515, 2014, doi: 10.1016/j.infsof.2013.11.003.
- [3] A. Rasnacic and S. Berzisa, 'Method for Adaptation and Implementation of Agile Project Management Methodology', *Procedia Comput. Sci.*, vol. 104, no. December 2016, pp. 43–50, 2016, doi: 10.1016/j.procs.2017.01.055.
- [4] A. Ahimbisibwe, R. Y. Cavana, and U. Daellenbach, 'A contingency fit model of critical success factors for software development projects', *J. Enterp. Inf. Manag.*, vol. 28, no. 2, pp. 14–39, 2015.
- [5] T. Dings, T. Dybo, and N. B. Moe, 'Agile Software Development : An Introduction and Overview', no. April, 2010, doi: 10.1007/978-3-642-12575-1.
- [6] C. Edeki, 'Agile software development methodology', *Eur. J. Math. Comput. Sci.*, vol. 2, no. 1, pp. 22–27, 2015.
- [7] S. M. Mohammad, 'DevOps automation and Agile methodology', *Int. J. Creat. Res. Thoughts*, vol. 5, no. 3, pp. 2320–2882, 2017, doi: 10.1729/Journal.24060.
- [8] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, 'Agile software development methods: Review and analysis', *VTT Publ.*, vol. abs/1709.0, 2002.
- [9] N. Ibrahim, 'An Overview of Agile Software Development Methodology and Its Relevance to Software Engineering', *J. Sist. Inf.*, pp. 69–80, 2007, [Online]. Available: www.agilealliance.org.
- [10] O. Sohaib and K. Khan, 'Integrating usability engineering and agile software development: A literature review', *2010 Int. Conf. Comput. Des. Appl. ICCDA 2010*, vol. 2, no. Iccda, pp. 32–38, 2010, doi: 10.1109/ICCDA.2010.5540916.
- [11] W. R. Fitriani, P. Rahayu, and D. I. Sensuse, 'Challenges in Agile Software Development : A Systematic Literature Review', 2016.
- [12] N. B. Moe, A. Aurum, and T. Dyba, 'Challenges of shared decision-making: A multiple case study of agile software development', *Inf. Softw. Technol.*, vol. 54, no. 8, pp. 853–865, 2012, doi: 10.1016/j.infsof.2011.11.006.

- [13] H. Hajjdiab and A. S. Taleb, 'Adopting Agile Software Development: Issues and Challenges Adopting Agile Software Development: Issues and Challenges', *Int. J. Manag. Value Supply Chain.*, vol. 2, no. September 2011, 2016, doi: 10.5121/ijmvsc.2011.2301.
- [14] R. Hoda and L. K. Murugesan, 'Multi-level agile project management challenges: A self-organizing team perspective', *J. Syst. Softw.*, vol. 117, pp. 245–257, 2016, doi: 10.1016/j.jss.2016.02.049.
- [15] K. Conboy, S. Coyle, X. Wang, and M. Pikkarainen, 'People over Process: Key Challenges in Agile Development', no. July 2015, 2011, doi: 10.1109/MS.2010.132.
- [16] J. Cho, 'Issues and Challenges of Agile Software Development with SCRUM', vol. IX, no. 2, pp. 188–195, 2008, [Online]. Available: https://doi.org/10.48009/2_iis_2008_188-195.
- [17] M. L. Drury-Grogan, K. Conboy, and T. Acton, 'Examining decision characteristics & challenges for agile software development', *J. Syst. Softw.*, vol. 131, pp. 248–265, 2017, doi: 10.1016/j.jss.2017.06.003.
- [18] P. Raj and P. Sinha, 'Project management in era of agile and devops methodologies', *Int. J. Sci. Technol. Res.*, vol. 9, no. 1, pp. 1024–1033, 2020.
- [19] L. Banica, M. Radulescu, D. Rosca, and A. Hagi, 'Is DevOps another Project Management Methodology?', *Inform. Econ.*, vol. 21, no. 3/2017, pp. 39–51, 2017, doi: 10.12948/issn14531305/21.3.2017.04.
- [20] A. Hemon, B. Lyonnet, F. Rowe, and B. Fitzgerald, 'From Agile to DevOps: Smart Skills and Collaborations', *Inf. Syst. Front.*, vol. 22, no. 4, pp. 927–945, 2020, doi: 10.1007/s10796-019-09905-1.
- [21] A. Dyck, R. Penners, and H. Lichter, 'Towards definitions for release engineering and DevOps', *Proc. - 3rd Int. Work. Release Eng. RELENG 2015*, no. May, p. 3, 2015, doi: 10.1109/RELENG.2015.10.
- [22] M. De Bayser, L. G. Azevedo, and R. Cerqueira, 'ResearchOps: The case for DevOps in scientific applications', *Proc. 2015 IFIP/IEEE Int. Symp. Integr. Netw. Manag. IM 2015*, no. June, pp. 1398–1404, 2015, doi: 10.1109/INM.2015.7140503.
- [23] S. I. Mohamed, 'DevOps Shifting Software Engineering Strategy Value Based Perspective', *IOSR J. Comput. Eng. Ver. IV*, vol. 17, no. 2, pp. 2278–661, 2015, doi: 10.9790/0661-17245157.
- [24] L. Riungu-Kalliosaari, S. Mäkinen, L. E. Lwakatare, J. Tiihonen, and T. Männistö, 'DevOps adoption benefits and challenges in practice: A case study', in *17th International Conference, Trondheim, Norway*, 2016, vol. 10027 LNCS, pp. 590–597, doi: 10.1007/978-3-319-49094-6_44.
- [25] by Patrick Debois *et al.*, 'Devops: A Software Revolution in the Making?', *J. Inf. Technol. Manag. Cut. IT J.*, vol. 24, no. 8, 2011, [Online]. Available: www.cutter.com.
- [26] A. J. Anderson, 'ScholarWorks Examination of Adoption Theory on the DevOps Practice of Continuous Delivery', 2019.
- [27] D. Spinellis, 'Being a DevOps Developer', *IEEE Softw.*, vol. 33, no. 3, pp. 4–5, 2016, doi: 10.1109/MS.2016.76.
- [28] A. Mishra and Z. Otaiwi, 'DevOps and software quality: A systematic mapping', *Comput. Sci. Rev.*, vol. 38, p. 100308, 2020, doi: 10.1016/j.cosrev.2020.100308.
- [29] S. S. Samarawickrama and I. Perera, 'Continuous scrum: A framework to enhance scrum with DevOps', *17th Int. Conf. Adv. ICT Emerg. Reg. ICTer 2017 - Proc.*, vol. 2018-Janua, no. March, pp. 19–25, 2017, doi: 10.1109/ICTER.2017.8257808.
- [30] G. Bou Ghantous, A. Gill, and G. Bou, 'DevOps: Concepts, Practices, Tools, Benefits and Challenges', in *Pacific Asia Conference on Information Systems (PACIS)*, 2017, p. 1, [Online]. Available: <http://aisel.aisnet.org/pacis2017/96>.
- [31] P. Perera, M. Bandara, and I. Perera, 'Evaluating the impact of DevOps practice in Sri Lankan software development organizations', *16th Int. Conf. Adv. ICT Emerg. Reg. ICTer 2016 - Conf. Proc.*, pp. 281–287, 2017, doi: 10.1109/ICTER.2016.7829932.
- [32] J. A. V. M. K. Jayakody and W. M. J. I. Wijayanayake, 'Challenges for adopting DevOps in information technology projects', in *2021 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, 2021, pp. 203–210.
- [33] N. Govil, M. Saurakhia, P. Agnihotri, S. Agarwal, and S. Sachin, 'Analyzing the Behaviour of Applying Agile Methodologies & DevOps Culture in e-Commerce Web Application', in *Fourth International Conference on Trends in Electronics and Informatics (ICOEI 2020)*, 2020, no. Icoei, pp. 899–902.
- [34] L. E. Lwakatare *et al.*, 'DevOps in practice: A multiple case study of five companies', *Inf. Softw. Technol.*, vol. 114, no. April, pp. 217–230, 2019, doi: 10.1016/j.infsof.2019.06.010.

- [35] L. Leite, C. Rocha, F. Kon, D. Milojcic, and P. Meirelles, 'A survey of DevOps concepts and challenges', *ACM Comput. Surv.*, vol. 52, no. November, 2019.
- [36] M. Fazal-Baqaie, B. Güldali, and S. Oberthür, 'Towards DevOps in multi-provider projects', *CEUR Workshop Proc.*, vol. 1806, pp. 18–21, 2017.
- [37] B. Chen, 'Improving the software logging practices in devops', in *2019 IEEE/ACM 41st International Conference on Software Engineering*, 2019, pp. 194–197, doi: 10.1109/ICSE-Companion.2019.00080.
- [38] A. Wahaballa, O. Wahballa, M. Abdellatif, H. Xiong, and Z. Qin, 'Toward unified DevOps model', *Proc. IEEE Int. Conf. Softw. Eng. Serv. Sci. ICSESS*, vol. 2015-Novem, no. September, pp. 211–214, 2015, doi: 10.1109/ICSESS.2015.7339039.
- [39] S. Rafi, W. Yu, and M. A. Akbar, 'Towards a Hypothetical Framework to Secure DevOps Adoption: Grounded Theory Approach', in *ACM International Conference Proceeding Series*, 2020, no. April, pp. 457–462, doi: 10.1145/3383219.3383285.
- [40] A. Wiedemann, N. Forsgren, M. Wiesche, H. Gewalt, and H. Krcmar, 'The DevOps phenomenon', *Res. Pract.*, vol. 17, no. 2, pp. 1–20, 2019, doi: 10.1145/3329781.3338532.
- [41] K. Kuusinen *et al.*, 'A large agile organization on its journey towards DevOps', *Proc. - 44th Euromicro Conf. Softw. Eng. Adv. Appl. SEAA 2018*, pp. 60–63, 2018, doi: 10.1109/SEAA.2018.00019.
- [42] M. M. Ahmad Ibrahim, S. M. Syed-Mohamad, and M. H. Husin, 'Managing quality assurance challenges of Devops through analytics', in *ACM International Conference Proceeding Series*, 2019, vol. Part F1479, no. January, pp. 194–198, doi: 10.1145/3316615.3316670.
- [43] A. Qumer Gill, A. Loumish, I. Riyat, and S. Han, 'DevOps for information management systems', *VINE J. Inf. Knowl. Manag. Syst.*, vol. 48, no. 1, pp. 122–139, 2018, doi: 10.1108/VJKMS-02-2017-0007.
- [44] M. Gokarna and R. Singh, 'DevOps: A historical review and future works', *Int. Conf. Comput. Commun. Intell. Syst.*, vol. 2001, pp. 2–7, 2020.
- [45] M. Munoz, M. Negrete, and J. Mejia, 'Proposal to avoid issues in the DevOps implementation: A Systematic Literature Review', in *Advances in Intelligent Systems and Computing (special issue from WorldCIST 2019 – 7th World Conference on Information Systems and Technologies)*, 2019, no. April, pp. 460–472, doi: 10.1007/978-3-030-16181-1.
- [46] S. Jones, J. Noppen, and F. Lettice, 'Management challenges for devops adoption within UK SMEs', in *QUDOS 2016 - Proceedings of the 2nd International Workshop on Quality-Aware DevOps, co-located with ISSTA 2016*, 2016, pp. 7–11, doi: 10.1145/2945408.2945410.
- [47] L. E. Lwakatare *et al.*, 'Towards DevOps in the embedded systems domain: Why is it so hard?', *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, vol. 2016-March, pp. 5437–5446, 2016, doi: 10.1109/HICSS.2016.671.
- [48] L. Yu Clemente Guerra, 'Exploring the disruptive power of adopting DevOps for software development', no. June, 2019, [Online]. Available: www.bth.se.
- [49] P. Perera, R. Silva, and I. Perera, 'Improve software quality through practicing DevOps', *17th Int. Conf. Adv. ICT Emerg. Reg. ICTer 2017 - Proc.*, vol. 2018-Janua, no. March, pp. 13–18, 2017, doi: 10.1109/ICTER.2017.8257807.
- [50] M. Rowse and J. Cohen, 'A Survey of DevOps in the South African Software Context', in *Proceedings of the 54th Hawaii International Conference on System Sciences*, 2021, vol. 0, pp. 6785–6794, doi: 10.24251/hicss.2021.814.
- [51] W. P. Luz, G. Pinto, and R. Bonifácio, 'Adopting DevOps in the real world: A theory, a model, and a case study', *J. Syst. Softw.*, vol. 157, pp. 1–16, 2019, doi: 10.1016/j.jss.2019.07.083.
- [52] B. B. Nicolau de França, H. Jeronimo, and G. H. Travassos, 'Characterizing DevOps by hearing multiple voices', in *ACM International Conference Proceeding Series*, 2016, pp. 53–62, doi: 10.1145/2973839.2973845.
- [53] P. Perera, M. Bandara, and I. Perera, 'Evaluating the impact of DevOps practice in Sri Lankan software development organizations', in *16th International Conference on Advances in ICT for Emerging Regions, ICTer 2016*, 2017, no. March 2018, pp. 281–287, doi: 10.1109/ICTER.2016.7829932.
- [54] M. D. Jose and Santos, 'The Definitive Guide to DevOps', *Project Management.com*, pp. 1–14, 2020.
- [55] I. Bucena and M. Kirikova, 'Simplifying the DevOps Adoption Process.', *Simpl. DevOps Adopt. Process*, 2017, [Online]. Available: <http://ceur-ws.org/Vol-1898/paper14.pdf>.

- [56] H. Myrbakken and R. Colomo-Palacios, 'DevSecOps: A multivocal literature review', *Commun. Comput. Inf. Sci.*, vol. 770, no. 1, pp. 17–29, 2017, doi: 10.1007/978-3-319-67383-7_2.
- [57] M. Stoyanova, 'Smart Concept for Project Management -Transition to DevOps', *Int. J. Knowl.*, vol. 34, no. 1, pp. 93–97, 2019.
- [58] M. Van Belzen, J. Trienekens, and R. Kusters, 'Critical success factors of continuous practices in a DevOps context', 2019.
- [59] T. Dumoulin, 'Critical Success Factors For DevOps Research Paper', 2017.
- [60] M. A. Akbar, S. Mahmood, M. Shafiq, A. Alsanad, A. A. A. Alsanad, and A. Gumaei, 'Identification and prioritization of DevOps success factors using fuzzy-AHP approach', *Soft Comput.*, vol. 0123456789, no. 2019, 2020, doi: 10.1007/s00500-020-05150-w.
- [61] Blueprint, 'Five Critical Factors for DevOps Success in Large , Complex Enterprises'.
- [62] V. Stray, N. B. Moe, and A. Aasheim, 'Dependency Management in Large-Scale Agile: A Case Study of DevOps Teams', in *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019, pp. 7007–7016, doi: 10.24251/hicss.2019.840.
- [63] Intland Software, 'The Top 5 DevOps Success Factors What is DevOps? Success Factors in Implementing DevOps'. 2021.
- [64] V. Robert, 'Critical Success Factors during the adoption and implementation of Continuous Delivery & Continuous Deployment in a DevOps context', 2020.
- [65] A. Wiedemann and M. Wiesche, 'Are you ready for Devops? Required skill set for Devops teams', *26th Eur. Conf. Inf. Syst. Beyond Digit. - Facet. Socio-Technical Chang. ECIS 2018*, 2018.
- [66] S. K. Duraisamy, B. Bass and S. Mukkavilli, 'Embedding performance testing in Agile Software Model', *International Journal of Software Engineering & Applications (IJSEA)*, vol. 12, no. 6, 2021, doi: 10.5121/ijsea.2021.12601
- [67] L. F. Amorim, M. Marinho and S. Sampaio, 'How (un)happiness impacts on software engineering in Agile teams?', *International Journal of Software Engineering & Applications (IJSEA)*, vol. 11, no. 3, 2020, doi: DOI: 10.5121/ijsea.2020.11303