LUT University

Understanding DevOps critical success factors and organizational practices

Azad Nasreen

This is a Publisher's version version of a publication

published by IEEE

in 2022 IEEE/ACM International Workshop on Software-Intensive Business (IWSiB)

DOI: 10.1145/3524614.3528627

Copyright of the original publication:

© 2022 Association for Computing Machinery

Please cite the publication as follows:

Nasreen Azad. 2022. Understanding DevOps critical success factors and organizational practices. In 5th International Workshop on Software-intensive Business: Towards Sustainable Software Business (IWSiB'22), May 18, 2022, Pittsburgh, PA, USA. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3524614.3528627

This is a parallel published version of an original publication. This version can differ from the original published article.

Understanding DevOps critical success factors and organizational practices

Nasreen Azad
nasren.azad@lut.fi
LUT University
Lappeenranta, Finland

ABSTRACT

DevOps is a combination of practices and a company culture which aims minimize the barriers between the operation and development teams in the organization. As its adoption and use in the industry has been growing, different kinds of research are trying to explore DevOps practices, processes and implementations in the organizations. Most of the extant research conducted in the past was to investigate how DevOps worked, what impacts it made into the organizations and how the adoption of DevOps played a role for the overall success for companies. This paper presents a qualitative analysis of a dataset collected via an open ended survey from 15 software professionals who are experienced in DevOps. The focus of the study is on reporting DevOps practices in organizations, and how they impact on the success of DevOps. We discuss the DevOps professionals point-of-view on DevOps practices and align their thoughts through a DevOps model of the critical success factors.

KEYWORDS

DevOps, Critical success factors, Software development, Continuous delivery, Continuous operation, Continuous integration, DevOps survey

ACM Reference Format:
Nasreen Azad. 2022. Understanding DevOps critical success factors and organizational practices. In *5th International Workshop on Software-intensive Business: Towards Sustainable Software Business (IWSiB'22), May 18, 2022, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3524614.3528627

1 INTRODUCTION

Companies are implementing DevOps and adopting the DevOps culture to improve the software delivery process [35]. DevOps minimizes the gaps between the operation and development teams once it is successfully implemented and adopted within the software organizations. The development process triggers the software deployment and that turns the software into production for the software organizations [11]. The main aspect of DevOps in an organization is to provide continuous delivery and continuous deployment so that the software is delivered faster with short delivery cycles [16].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
IWSiB'22 , May 18, 2022, Pittsburgh, PA, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9302-7/22/05…$15.00
https://doi.org/10.1145/3524614.3528627

For companies, implementation and adoption of DevOps practices is a popular phenomena now. According to an annual report published on 'State of DevOps' shows that there is an increase in DevOps use in Europe, approximately 29 percent in 2019 and in 2021 the percentage has increased into 32 percent [17].

DevOps is a well established culture in companies now [30]. DevOps is considered as an organizational culture because DevOps acts as a cultural shift towards collaboration (Development, operations and quality assurance) [15].Many studies were conducted on the DevOps process, implementation, adoption and challenges but there is a lack of empirical research on DevOps to develop and validate models of critical success factors. Most prior studies done in DevOps are case studies which explains DevOps practices in the organization [13, 25, 28]. Therefore, current studies fall short in explaining what are the *critical success factors*, and how they impact on DevOps projects in organizations.

In this paper we will investigate these issues based on an online survey with open-ended questions among DevOps professionals from different companies. Consequently, we address the following research question in this paper:

RQ: What are the critical success factors of DevOps projects as reported by the professionals?

In order to address the above research question, we adopted the DevOps critical success factor model proposed by [2] and collected data among 15 DevOps professionals through an online survey with open-ended questions. We conducted a qualitative analysis on the collected data to test the proposed critical success factor model. Our findings confirm that the success factors can be of three types as discussed by [2] namely technical, organizational, and social and cultural factors. This in-progress study contributes to the growing literature of DevOps research by empirically supporting the Devops critical success factor model.

The remaining of this study is organized as follows. Section 2 presents the central concepts, DevOps as well as Critical Success Factory theory, and their related literature. It is followed by the description of the empirical data 40 collection and the research process in Section 3. Section 4 presents the results, Section 5 discusses on their impacts, and Section 6 concludes the study.

2 BACKGROUND

2.1 DevOps concepts and process

In recent times the software development continuous process requires to accommodate faster and quicker release of software. For this purpose the end user feedback is very crucial and the faster the feedback is given by the end user the better the performance of the software [24]. To increase the capability of the software delivery

with an automated manner, it is important to implement and adopt DevOps [20]. DevOps critical success factors the major indicators of a successful implementation, adoption for DevOps process [2]. In this section we will discuss DevOps concept, process and critical success factors' impacts on the overall DevOps practices in organizations. For our literature review, we have used empirical studies which specifically focused on the DevOps concepts and practices in organizations.

The business environment of today's world is changing rapidly. To keep pace with this dynamic environment, it is essential to grow with the flow. This change also impacts the software development process. Customer demands and expectations are higher and they expect that the software applications should be provided with higher quality with a quicker delivery time [6]. Various methods were used by companies for software development. Among those agile practices are used widely. The problem with agile practices is it only works for the development sides. A software process needs both the development and operations teams' combined effort, otherwise the process is incomplete [21]. To get an adequate business value it is important that the software should be delivered to the production environment, otherwise even if the software development process was faster it does not add any value to the business [21].

DevOps increases the performance of software products and services and ensures the collaboration, integration and communication process [3]. There are many concepts surrounding DevOps but there is no specific DevOps definition to follow [22]. The purpose of DevOps is to eliminate the gap among teams so that the software development can be done efficiently [38]. The development teams deliver the software features more frequently but the operations teams can not implement the features rapidly. Thus, the main advantage DevOps serves here is delivering faster features to customers. DevOps merges the development teams and operation teams to work together to achieve the faster software development process [38]. Jabbari et al [22] have conducted a comprehensive content analysis and found eight components of DevOps and defined DevOps as a development methodology which eliminates the gap between Dev and ops teams. They also suggested that communication and collaboration, continuous integration, delivery with automated deployment, ensuring quality assurance can be achieved by DevOps.

In companies the working practice of the operations team and development teams are different. Many times the alignment and working process vary and it is not surprising that there might be a conflict between Dev and Ops teams. For example, when the the development team is working with the faster delivery, the operation team observes the stability of the system, meaning both teams are working in different stages of the development process. This kind of situation decreases the work stability and makes the software development process slower. These conflicting situations can be handled with DevOps adoption in organizations [21]. The DevOps practices include synergistic collaboration, shared goal, shared working space with common work alignment, and team trust. This can create better collaboration and understanding between teams and that will contribute to a better software development process [6]. Another driving force for DevOps adoption includes automation as it helps to utilize less resources for software development process maintenance [6].

2.2 DevOps Critical Success Factors

The management literature discussed the critical success factors at the beginning of the 1960s [2, 14]. There has been many literature on critical success factors, and related to their impacts and roles in the organization. Critical success factors can be explained as *"a key areas of processes where acceptable results are required for specific managers to achieve the goal"* [7]. According to Azad and Hyrynsalmi [2], there are various critical success factors for DevOps projects discussed in prior studies. According to Shahin et al. [31] when practitioners are aware of the critical success factors beforehand, it will help the organization to handle continuous practices and smooth integration of DevOps process.

Prior literature suggest how performance engineering complexity can be addressed for DevOps projects success. According to Bezemer et al. performance engineering approaches should be lightweight to address DevOps performance in the organization [5]. Performance measurement [34], Software release and [8] Seamless upgrades [10] are also the important factors. According to Chen et al. [9] performance regression needs to be identified at the beginning of the source code level so that DevOps operations can run smoothly. According to Claps et al. [10] continuous integration process and support has an impact on DevOps success. Furthermore, automatic testing technique improvement [27], test automation [27], Verification of soundness [4], automation tools [4] may affect the success of DevOps.

Literature also suggests that team and process efficiency have some impacts on DevOps success [12]. Smite et al [32] suggested that team efficiency is hampered by reluctant to ask questions to superiors and lack of understanding of the fact. The study suggested that DevOps practices in Offshore teams might be reluctant to voice criticism to receive guidance from the superiors. According to Akbar et al. [1], cross functional teams and the teams with DevOps skills impact the success of the organization. Tsanos et al. [34]. and Kolfschoten [23] claimed that mutuality, trust, commitment and mutual respect among teams impact the team efficiency for DevOps.

Yoon et al [37] conducted a survey and found that clear role setting, methods to solve issues in teams are very crucial. Wettinger et al. [36] stated that team collaboration is important for DevOps success. Kolfschoten et al. [23] suggested that the teams efficiency increases when team mates understand each other and respect each other's opinion. Nybom et al. [26] found from survey that collaboration, trust, smoother workflow, shared responsibilities, and common ways of working increases the team's efficiency. Claps et al. [10] stated that team roles, teams coordination, team experience, source code control has an impact on DevOps success.

Riungu et al. [27] suggested that cultural shifts have an impact on the overall objective of the organization. It is important to have a common goal for the teams to achieve DevOps adoption in the company. According to Akber et al. [1] emphasizing culture is more important than emphasizing the tools for task execution in teams. Moreover cultural norms [23]and organizational cultures [26]are vital issues to consider while doing team work.

Azad and Hyrynsalmi [2] proposed a synthesized framework for the critical success factors of DevOps projects. According to them, there are ten major factors which are categorized as technical,

organizational, social and cultural factor groups. The framework is shown in Figure 1. In this model, we suggest that technical factors and organizational factors (intraorganizational collaboration, organization hierarchy, and strategic planning) would directly impact DevOps success. The social and cultural factors (team dynamics and cultural shift) would moderate the effects of organizational factors on DevOps success.In addition, social and cultural factors might also impact DevOps success directly. In addition, Azad and Hyrynsalmi [2] hypothesize that. However, their model is based on synthesizing extant scientific literature on DevOps and it have not been thus far empirically evaluated or supported. In this paper, we adopt and aim to provide first empirical verification for this model.

3 RESEARCH APPROACH

3.1 Data collection

Schluter et al. [29] suggest that an open-ended survey questionnaire is a relevant method for collecting data from a large number of respondents. Following this, we have collected the data by using open ended survey questionnaires from a specific group of respondents: the DevOps professionals.

For collecting data we have conducted an online survey. An open ended questionnaire was developed by using Google Form. A total of 25 questions were developed to be included in the survey. Among those questions, 20 questions include themes related to DevOps process, DevOps adoption, DevOps implementation and DevOps challenges. These questions were open-ended. The respondents were encouraged to write and give reasons for their answers.Rest of the five questions include demographic questions regarding country of origin, role of the respondents, size of the company, and experience of professionals in software development process.

We shared our survey link through different social media platforms such as Facebook, LinkedIn, and Twitter. The questionnaire was opened in the end of November 2021 and it was kept open for one month. Due to the online open invitation nature of the survey, it is hard to estimate how many people the survey reached. Yet, for example the Twitter messages were seen by at least 1,117 tweeters and the Linked posts by at least 894 readers.

We received 15 usable responses from the DevOps professionals. The survey responses were from Finland, Netherlands, Spain, Sweden, Trinidad and Tobago and United kingdom. The size of the company they were representing ranged from 50-20,000 employees. The role of the respondents in the company were Head of technology, Tech lead, Scrum Master, Site Reliability Engineer, DevOps engineer, Software specialist, Business Analyst, Could Engineer, Technical Project manager and software Engineer. The respondents' experience in the software development ranged from 3 to 20 years. Most respondents reported their in-depth experience with DevOps. Thus, after merging the responses, we had approximately 30 pages of texts data for analysis purposes.

3.2 Data analysis

We have content analyzed the collected data [18]. We have classified the reported critical success factors in three broad categories with various subcategories as described by [2]. This type of data analysis technique is known as directed content analysis. Directed content analysis is a technique which is used to validate and extend a theoretical framework [19].

Two researchers were involved in coding and categorizing the data. At first the researchers coded and categorized the survey data separately. The process was the combination of repeated and careful readings. Then the data was sorted and divided into three main categories: Organizational, Technical, Social and cultural factors. While doing the categories the researchers put sub categories based on themes of the texts. The author of this paper, who is a doctoral student, is one of the coders. After completing the coding process the researchers compared their views with each other. There were some agreements and disagreements. The disagreements were solved by discussion. Most of the disagreements were based on the social and cultural factors and organizational success factors. Some of the respondents' answers could be categorized both as cultural factors and organizational factors. For example, the team dynamics and cultural shifts responses were kind of similar with Intra organizational collaboration and communication and strategic planning.

4 RESULTS

This section presents the findings of our data analysis which identifies similar success factors that have been discussed in the prior literature. Our findings also shed lights on DevOps generalized practices in various organizations by analyzing the viewpoints of DevOps professionals. We have found that the identified critical success factors are directly linked with DevOps organizational practices.

The results of our study are presented according to three dimensions — organizational, technical as well as social and cultural — as identified by [2] in their literature survey. We will discuss three main categories and their factors in regards to practical issues and practices from DevOps professionals' point-of-views.

4.1 Organizational success factors

In DevOps success the organizational success factors play an important role. According to Azad and Hyrynsalmi [2]'s model, the organizational success factors has three sub categories. Those sub categories include intra-organizational collaboration or communication, organizational hierarchy and strategic planning.

4.1.1 Intra-organizational collaboration or communication. Intra-organizational collaboration or communication is the subcategory of organizational success factors. Our data analysis shows that collaboration between the Dev and ops team is essential for DevOps success in the organization. It is very essential to have shared responsibilities among dev and ops teams. Shared responsibilities might also lead to effective communications among the teams. Miscommunication between the Dev and Ops team is a major barrier for the DevOps process success.

For example, the respondents reported that:

> "In our organization, DevOps (Development and Operations) is a discipline that stresses communication, collaboration, integration and even fusion between application development and operations with the goal of promoting fast, consistent, high-quality, scalable

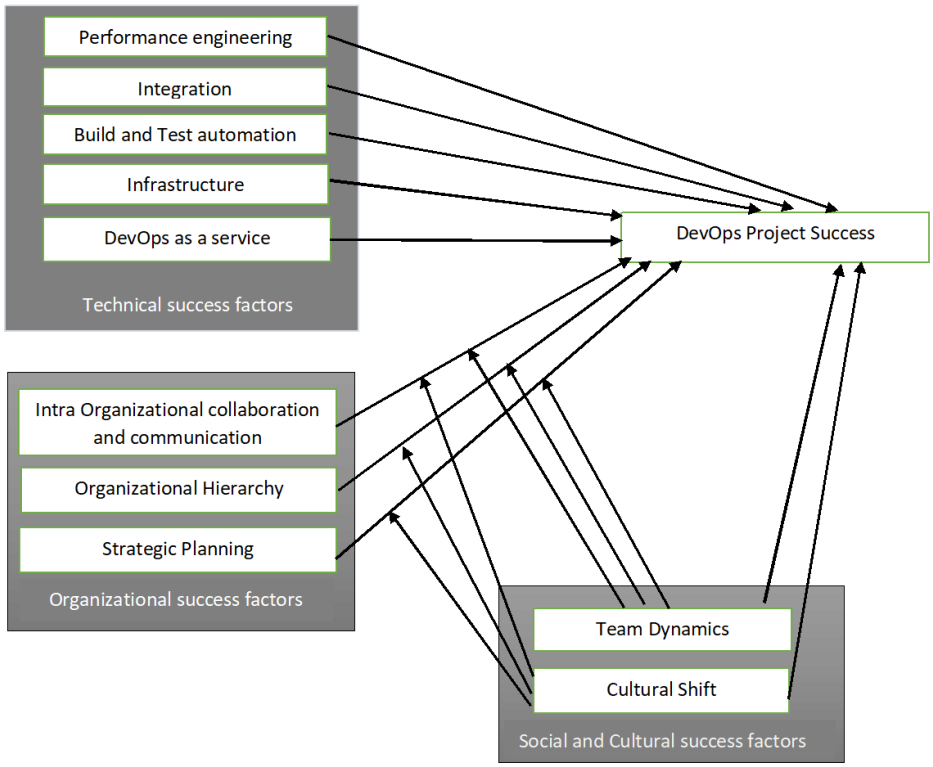

Figure 1: The DevOps Critical Success Factor model (adapted from [2]).

application releases. It is a method of bringing developers, IT operations and Quality Assurance (QA) team together".

"In my current project, we have 2/3 multi-site developer teams, 2/3 teams for testing (integration, user-experience,...). Usually, members from development and test teams share the responsibility of build environment. Product management takes care of configuration and release management. A separate incident management team takes care of the bugs and issues in software".

4.1.2 Organizational hierarchy. Organizational hierarchy is another subcategory of organizational success factors. From our data analysis, we have observed that the DevOps professionals are willing to get instructions from superiors. It is vital to get instructions from managers or lead persons who could give approval for different team tasks. Without a lead person, the team work might not not run smoothly. For example, some of the respondent reported that:

"...To prioritize tasks the team meets periodically. Product management assign priority level on high level business requirements based on the market demand and prospects of future. Production should run without any problem. Development has second priority after smooth Production run. Project leads make decisions on prioritization and make the decision for the team...."

"...The product owner and team lead prioritize and take the decision based on existing issues and need. It also depends on the sprint planning and requirements for the upcoming release of the software development..."

4.1.3 Strategic Planning. Strategic planning is the last category. Our findings show that from a strategic point of view, the product management team maintains the business requirements and the technology team handles the technical requirements based on the availability of resources and business needs. Therefore, planning related to initiatives, resources, and budget are necessary for the overall software process. The IT and the business plan should have similar goals and objectives. Continuous delivery and continuous development could ensure the quality of the product. Therefore strategic planning should take into account company pressure, meeting deadlines, change management, reduce the time to market Azad and Hyrynsalmi [2].

For example, a few of the answers were:

"..The product management team is responsible for the functional (business) requirement and the technology

team (Technology Architects) are responsible for taking non-functional (technical) requirements based on the business requirements and available resources.."'.

"...Continuous Integration, Continuous Testing, Continuous Delivery, Continuous Deployment are the areas where my organization implements modern DevOps practice..."

"...Project manager/ Scrum Master coordinates with the equivalent role on the client's side to determine availability of resources and prioritize requirements based on that..."

"... More optimized production process, better stronger team producing high quality products, Collaboration and work in multiple disciplines of software life cycle, DevOps helped in faster and cheaper delivery of software products, increasing velocity of change and time to market, empowering people in their work, autonomous teams, and less downtime due to more controlled changes are positive impacts to name a few".

4.2 Social and cultural factors

Social and cultural factors are one of the leading factors for DevOps success. According to [2], social and cultural factors have two subcategories named team dynamics and cultural shift. Below we discuss our findings on these two subcategories.

4.2.1 Team Dynamics. Team Dynamics is essential for DevOps success. When the professionals work in a team, they need to share knowledge and ideas with the team members. This in turn help the team members to improve the work efficiency. Team dynamics benefits the DevOps team to improve collaboration, trust building, better workflow and common ways of working in the team. Team work can help in faster software delivery, increased quality of the software, early detection of defects, and increased security. The respondents also mentioned that a typical team consists of DevOps Lead/Architect, a few junior-mid-senior developers with similar kind of skill set.

For example, some respondents reported on the following issues:

"Sharing the knowledge and creating a diversified culture helps the team to progress better, tech lead is responsible for designing the architecture and confirm the architectural structure".

"Product owners are responsible for specification along with the team members and sometimes the stakeholders and lawmakers can have an impact on teams for doing certain works".

"Faster delivery times, increased quality, early defect detection, improved security and happier developers are our team dynamics".

"Knowledge sharing in teams, team inspiration, delivers better and faster feature.We try to share knowledge to other teams as well which is a good team culture".

"Usually the team consists of DevOps Lead/Architect, a few Junior-Mid-Senior Developers with similar kind of skill set. Lead developer usually distributes the work among the other team members and provides the guidelines for expected outcomes. ".

4.2.2 Cultural shift. Cultural shift has a huge impact on DevOps success in organizations. In an organization, there are barriers related to technical issues however, the barriers related to the cultural issues are even more important while adopting DevOps. From the respondents' answers, we have found that organizational culture plays a vital role for achieving objective of the project. Sharing the experience and creating a diversified culture helps the team to progress better.

The companies should emphasize on culture rather than emphasizing tools. This will help the company to execute the tasks quicker. DevOps transformation, good practices, understanding the cultural norms are essential for the DevOps teams. Some respondents reported the cultural issues as discussed below.

"One DevOps Center of Excellence team tries to spread the word and practices, when rest of the old fashion company tries to hide from it and the most agile teams and units trying to survive in the middle of DevOps transformation by taking good practices in use as much is possible with mix of cloud and on-prem environments. Most of the time documented guidelines doesn't work like they are given to the teams".

"Employees joins and also leaves. So for a small team like us we manage our CI/CD pipeline ourselves by sharing the knowledge and creating a diversified culture".

"Teams work on agile methods to come up with requirements and features. I'd say we have more or less shared responsibilities for these. Culture emphasizes making development operations easier and faster".

"Culture and mindset in the organizations and teams is supportive and engaged in what each other is doing. Also like to have fun and joke around about the client situations we have to deal with".

"Team collaboration helps us to build awesome products for clients. The company culture we have is Collaborative, Flexible, Supportive, Focused, Goal Oriented, Can do attitude. Our company has Very well adopted DevOps culture in every stages of the development life cycles".

4.3 Technical factors

Technical success factors are another category which have five sub categories named 'Performance engineering', 'Integration', 'Build and test automation', 'Infrastructure', and 'DevOps as a service'[2]. Below we will discuss the findings from our study and how that can be related to the technical success factors and their sub categories.

4.3.1 Performance engineering. For any software, performance engineering is a complex activity. It requires adequate knowledge to understand the software life cycle. Performance engineering

approaches are mostly complex and not easy to adopt for the professionals. According to Bezmer et al. [5] the application of performance models is rare in industry due to lack of participants' knowledge on performance modeling.

In DevOps there is a low adoption of performance engineering. The reason behind low adoption is that the performance engineering approaches are not designed for DevOps context. There are many existing tools that are used for DevOps settings in organizations, for example Docker and Puppet but they do not align smoothly with the performance engineering process [5]. Because of that many professionals still depend on SSH and system tools while working for software deployment and performance monitoring. Many professionals avoid using specialized tools such as APM (Application performance monitoring) tools.

Performance engineering approaches should be lightweight. Performance engineering tools should integrate properly with the existing DevOps pipelines otherwise the system will not work. We did not observe direct evidences related to performance engineering in our data. However, some respondents mentioned (see below) about their views on DevOps pipelines' execution time and necessity of related tools.

> "CI/CD pipeline execution time which is related to the developers work to mark a ticket ready, tasks prevent the pipeline to be executed properly, customer support needs, e.g. a temporary review site to showcase the provided solution, version discrepancy that halts new feature to be released.".

> "There are various tools for supporting development depending on different programming languages. In our workplace, common tools are IntelliJ IDE, GIT, Gitlab, Bash scripting, aws toolkit, MongoDB tools etc".

4.3.2 Integration. Continuous integration tools do several automated actions which help the system to work together for the pipeline. Some of the pipeline stages include package generation, automated test execution, code verification, and deployment for the production and development environments. The developers are the responsible actors for defining pipeline structures. On the other hand, operators are responsible for defining collaboration for deployment phases. They are also responsible for the continuous integration. In the survey data some respondents reported that for continuous integration the developers use Jenkins or GitLab CI as an integration tool. For example, some respondents reported that:

> "Usually, developers are responsible of development and unit test. Then the Test team is responsible for integration test, user acceptance test and E2E tests. After this, the quality matrix is analyzed by the product management to provide feedback and make decision on release".

> "For the integration process our company uses various tools. Some of the tools includes Version control (Git), dependency management tools for programming languages, testing frameworks, container technologies, Continuous Integration and Delivery tools (CI/CD), cloud platforms, configuration management tools, infrastructure as code tools".

> "Linux (Operating System), Git/GitHub/BitBucket (Source repository), Jenkins (Build), JIRA (Project requirement, task and incident management), SonarQube (Test, Quality Assurance), Docker, AWS and Azure (Deployment) are used as a integration tool".

> "Make the functional requirement at a considerable large number (granular level of tasks/sub-tasks) and release the task/sub-task to the project source repository when it is completed and unit tested instead of releasing in one-go. Too much code to release at one time makes integration and debugging messy if something goes wrong".

4.3.3 Build and Test automation. Build and test automation is another sub category of the technical success factors. Based on our data analysis, we have found that test automation, automatic testing, improvement of techniques might impact the success of DevOps. In teams different automation techniques are used. Some of the automation includes verification of soundness, automation specific tools, build and process automation etc. Some of the respondents reported that:

> "Depending on the project and organization, changes could be deployed to production instantly after merging them to the main branch. This can be done using automation.The process mentioned above can also be applied for infrastructure and operations when Infrastructure as Code (IaC) and config management tools are used. So all in all, the process is rather similar to agile software development".

> "I found mostly positive impacts by introducing DevOps in our organizations. For example, test automation, easy and automatic delivery to different tracks etc.Ease and speed in delivering software to the customer, Fast and reliable releasing, test automation, Faster Software development process, Better Security practices, Better quality, Opportunity to explore many tools and technologies".

> "The automated DevOps help to progress software development in a faster way, reduce lot of manual work and it helps detect issues at very early stages of the development. Also, it is very helpful for release management and deployment to different environment (DEV, STAGE, PRODUCTION) automatically".

4.3.4 Infrastructure. For software process integration the infrastructure play a crucial role. Infrastructure helps the DevOps teams to handle the software and hardware issues and make sure the software to be deployed when required. For a better execution of DevOps practices in the organization, it is necessary to have infrastructure provisioning and infrastructure development. Some of the view points of the respondents are given below.

> "Depending on the project, there could be a central "DevOps" team which provides services, tooling, and

> support for software development teams. Or, the software development team(s) can manage everything by themselves".
>
> "Mainly one dedicated team who are responsible for managing the overall infrastructure and DevOps activities. However, other development teams, responsible for developing software solutions, also have resources who have knowledge about the DevOps activities".

4.3.5 DevOps as a Service. Micro-services are part of the cloud infrastructure. It is essential for future DevOps development. Because microservices can enable faster deployments in the cloud without hampering the performance of the software process. DevOps adoption ideology suggests that separate software teams are responsible for the different stages of software applications and allow the team to develop, test, handle failures and act independently [33].

Microservices are essential part of DevOps practices. DevOps teams works together to maintain the life cycle process through continuous delivery and continuous release. We have found a response in regards to micro services. The respondents stated that:

> "The "Ops/Infra/Cloud/whatever" team owns and maintains the general shared infra between all microservices etc. plus offers support whenever needed. I'm part of the Ops team and we work closely with developers pretty much every day".

5 DISCUSSION

5.1 Findings

This paper addresses the question of what are the critical success factors of DevOps projects as mentioned by the professionals. From the analyzed responses to the open-ended questions—filling 30 pages of text—by 15 DevOps professionals, there are reported success factors which can be categorized into organizational, social & cultural, technical categories. While the actual impact of the reported success factors cannot be evaluated due to their importance of being perceived and mentioned by the respondents, the reported factors are rather well-aligned with those reported in the extant academic literature.

While more support were found for organizational and social & cultural factors, also technical factors were present in the data. However, the technical problems and solutions are quite specific and not that often reported. Yet, there are exceptions as, for example, there are several issues belonging into Build and Test automation technical factor. Furthermore, it is worth to note that in all organizations—whether producing software or not—organizational, cultural and social aspects are present when new practices and processes are taken into use. Therefore, their presence in the list of the critical success factors of DevOps usage is not a surprise.

Overall, this in-progress study offer empirical support for the DevOps critifal success factors model synthesized from the literature by [2]. Issues belonging into each factors were found in the analysis. Furthermore, no new factors—that were not present in the model—were identified. However, the number of respondents remained small and new critical success factors could be identified with more qualitative, e.g. a case study research, approach. In addition, in the further studies the model might need updates on, e.g., the technical factors whether they are frequently present in DevOps projects or not.

5.2 Research limitations and future work

There are certain noteworthy limitations. First, the data was collected through an open-invitation online survey. It indeed helped us reach more respondents, however, interviews would have provided richer dataset and new angles into the phenomenon at hand. Therefore, we plan to conduct interviews with our respondents in the future.

Second, identifying respondents who had expertise on DevOps was a huge challenge. As DevOps is still a progressing practice for many companies, finding respondents was difficult and this is one of the major limitations of this research. Third, some respondents had lack of knowledge of DevOps as they were new to the company and were introduced to DevOps shortly. Some of the employees have joined to the company recently and not aware of the process and culture. They were unable to answer some specific questions regarding team dynamics and team coordination issues. Fourth, in this research there could be a selection bias and response bias. Some answers of the respondents were not specific and there could be some confidential issues related to sharing internal information.

Nevertheless, this study showed some empirical support for the DevOps critical success factor model and its component. However, more work is needed to both develop the model further as well as empirically validate its components. In addition, while the model hypothesize that there are some interlinked effects, as illustrated in Figure 1, their interlinked relationships were not addressed in this study and further work is needed to understand those.

Moreover the author is aiming to structuring the CFS based on the DevOps principles, DevOps process, challenges and adoption.

6 CONCLUSIONS

DevOps is a culture which is changing the way development teams and operations teams operate in today's world. DevOps culture in a company requires to prioritize various issues, implement new tools, acquire new knowledge and creative mindset for the DevOps professionals. The critical success factors of DevOps play a significant role in successful DevOps practices in the companies. Critical success factors can help the companies to change the current culture and will support the future changes in the DevOps practices.

In this paper, we have discussed the viewpoints of DevOps professionals on critical success factors and how that impact the DevOps process, adoption and implementation. We have provided empirical support to validated the DevOps critical success factor model based on the DevOps professionals data. We have shown that there are three major categories of success factors named Technical success factors, Social and Cultural success factors and organizational success factors. The model can be used by the researchers and practitioners to understand the success factors that might affect DevOps in the organizations. The IT professionals will be benefited with this paper as it gives a generalized views from different Devops company professionals regarding DevOps process, success factors, DevOps adoption, and DevOps implementation.

ACKNOWLEDGMENTS

This research was conducted by a doctoral student for her PhD dissertation. The author would like to give thanks to all the DevOps professionals who have participated in the survey and provided meaningful responses for conducting the research.

REFERENCES

[1] Muhammad Azeem Akbar, Sajjad Mahmood, Muhammad Shafiq, Ahmed Alsanad, Abeer Abdul-Aziz Alsanad, and Abdu Gumaei. 2020. Identification and prioritization of DevOps success factors using fuzzy-AHP approach. *Soft Computing* (2020), 1–25.

[2] Nasreen Azad and Sami Hyrynsalmi. 2021. What Are Critical Success Factors of DevOps Projects? A Systematic Literature Review. In *Software Business. 12th International Conference, ICSOB 2021, Drammen, Norway, December 2–3, 2021, Proceedings*, Xiaofeng Wang, Antonio Martini, Anh Nguyen-Duc, and Viktoria Stray (Eds.). Springer, 221–237.

[3] Len Bass, Ingo Weber, and Liming Zhu. 2015. *DevOps: A software architect's perspective*. Addison-Wesley Professional.

[4] Walid Ben Mesmia, Mohamed Escheikh, and Kamel Barkaoui. 2021. DevOps workflow verification and duration prediction using non-Markovian stochastic Petri nets. *Journal of Software: Evolution and Process* 33, 3 (2021), e2329.

[5] Cor-Paul Bezemer, Simon Eismann, Vincenzo Ferme, Johannes Grohmann, Robert Heinrich, Pooyan Jamshidi, Weiyi Shang, André van Hoorn, Monica Villavicencio, Jürgen Walter, et al. 2019. How is performance addressed in DevOps?. In *Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering*. 45–50.

[6] Ineta Bucena and Marite Kirikova. 2017. Simplifying the DevOps Adoption Process.. In *BIR Workshops*. 1–15.

[7] Christine V. Bullen and John F. Rockart. 1981. *A primer on critical success factors*. Working papers 1220-81. Report (Alfred P. Sloan School of Management. Center for Information Systems Research) ; no. 69. Massachusetts Institute of Technology (MIT), Sloan School of Management. https://EconPapers.repec.org/RePEc:mit:sloanp:1988

[8] Matt Callanan and Alexandra Spillane. 2016. DevOps: making it easy to do the right thing. *Ieee Software* 33, 3 (2016), 53–59.

[9] Jinfu Chen. 2020. Performance regression detection in DevOps. In *2020 IEEE/ACM 42nd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*. IEEE, 206–209.

[10] Gerry Gerard Claps, Richard Berntsson Svensson, and Aybüke Aurum. 2015. On the journey to continuous deployment: Technical and social challenges along the way. *Information and Software technology* 57 (2015), 21–31.

[11] Patrick Debois et al. 2011. Devops: A software revolution in the making. *Journal of Information Technology Management* 24, 8 (2011), 3–39.

[12] Jessica Díaz, Rubén Almaraz, Jennifer Pérez, and Juan Garbajosa. 2018. DevOps in practice: an exploratory case study. In *Proceedings of the 19th International Conference on Agile Software Development: Companion*. 1–3.

[13] Jessica Díaz, Jorge E Perez, Agustin Yague, Andrea Villegas, and Antonio de Antona. 2019. Devops in practice–a preliminary analysis of two multinational companies. In *International Conference on Product-Focused Software Process Improvement*. Springer, 323–330.

[14] Roger A Dickinson, Charles R Ferguson, and Sumit Sircar. 1984. Critical success factors and small business. *American Journal of Small Business* 8, 3 (1984), 49–57.

[15] Christof Ebert, Gorka Gallardo, Josune Hernantes, and Nicolas Serrano. 2016. DevOps. *Ieee Software* 33, 3 (2016), 94–100.

[16] Brian Fitzgerald and Klaas-Jan Stol. 2017. Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software* 123 (2017), 176–189.

[17] Nicole Forsgren, Dustin Smith, Jez Humble, and Jessie Frazelle. 2019. 2019 accelerate state of devops report. (2019).

[18] Ole R Holsti. 1969. Content analysis for the social sciences and humanities. *Reading. MA: Addison-Wesley (content analysis)* (1969).

[19] Hsiu-Fang Hsieh and Sarah E Shannon. 2005. Three approaches to qualitative content analysis. *Qualitative health research* 15, 9 (2005), 1277–1288.

[20] Jez Humble and Joanne Molesky. 2011. Why enterprises must adopt devops to enable continuous delivery. *Cutter IT Journal* 24, 8 (2011), 6.

[21] M Huttermann. 2012. DevOps for Developers: Integrate Development and Operations. *The Agile Way* (2012).

[22] Ramtin Jabbari, Nauman bin Ali, Kai Petersen, and Binish Tanveer. 2016. What is DevOps? A systematic mapping study on definitions and practices. In *Proceedings of the Scientific Workshop Proceedings of XP2016*. 1–11.

[23] Gwendolyn L Kolfschoten, Gert-Jan de Vreede, Robert O Briggs, and Henk G Sol. 2010. Collaboration 'engineerability'. *Group Decision and Negotiation* 19, 3 (2010), 301–321.

[24] Mika V Mäntylä, Bram Adams, Foutse Khomh, Emelie Engström, and Kai Petersen. 2015. On rapid releases and software testing: a case study and a semi-systematic literature review. *Empirical Software Engineering* 20, 5 (2015), 1384–1425.

[25] Carl Marnewick and Josef Langerman. 2020. DevOps and organisational performance: the fallacy of chasing maturity. *IEEE Software* (2020).

[26] Kristian Nybom, Jens Smeds, and Ivan Porres. 2016. On the impact of mixing responsibilities between devs and ops. In *International Conference on Agile Software Development*. Springer, 131–143.

[27] Leah Riungu-Kalliosaari, Simo Mäkinen, Lucy Ellen Lwakatare, Juha Tiihonen, and Tomi Männistö. 2016. DevOps adoption benefits and challenges in practice: a case study. In *International conference on product-focused software process improvement*. Springer, 590–597.

[28] James Roche. 2013. Adopting DevOps practices in quality assurance. *Commun. ACM* 56, 11 (2013), 38–43.

[29] Jessica Schluter, Philippa Seaton, and Wendy Chaboyer. 2008. Critical incident technique: a user's guide for nurse researchers. *Journal of advanced nursing* 61, 1 (2008), 107–114.

[30] Mark Schwartz. 2017. *A Seat at the Table and The Art of Business Value*. It Revolution.

[31] Mojtaba Shahin, Muhammad Ali Babar, Mansooreh Zahedi, and Liming Zhu. 2017. Beyond continuous delivery: an empirical investigation of continuous deployment challenges. In *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE, 111–120.

[32] Darja Šmite, Nils Brede Moe, and Javier Gonzalez-Huerta. 2021. Overcoming cultural barriers to being agile in distributed teams. *Information and Software Technology* 138 (2021), 106612.

[33] Demetris Trihinas, Athanasios Tryfonos, Marios D Dikaiakos, and George Pallis. 2018. Devops as a service: Pushing the boundaries of microservice adoption. *IEEE Internet Computing* 22, 3 (2018), 65–71.

[34] Christos S Tsanos, Konstantinos G Zografos, and Alan Harrison. 2014. Developing a conceptual model for examining the supply chain relationships between behavioural antecedents of collaboration, integration and performance. *The International Journal of Logistics Management* (2014).

[35] N Forsgren Velasquez, Gene Kim, Nigel Kersten, and Jez Humble. 2014. State of DevOps report.

[36] Johannes Wettinger, Uwe Breitenbücher, and Frank Leymann. 2014. Devopslang–bridging the gap between development and operations. In *European Conference on Service-Oriented and Cloud Computing*. Springer, 108–122.

[37] Changbyung Yoon, Keeeun Lee, Byungun Yoon, and Omar Toulan. 2017. Typology and success factors of collaboration for sustainable growth in the IT service industry. *Sustainability* 9, 11 (2017).

[38] Mohammad Zarour, Norah Alhammad, Mamdouh Alenezi, and Khalid Alsarayrah. 2019. A research on DevOps maturity models. *Int. J. Recent Technol. Eng* 8, 3 (2019), 4854–4862.