AN ANALYSIS OF DEVOPS' IMPACT ON INFORMATION TECHNOLOGY ORGANISATIONS: A CASE STUDY

A. Mudadi[1*] & H.H. Lotriet[1]

ARTICLE INFO

Article details
Submitted by authors 17 Jul 2022
Accepted for publication 9 Apr 2023
Available online 26 May 2023

Contact details
* Corresponding author
austinmdd@gmail.com

Author affiliations
1 Department of Information Systems, University of South Africa, South Africa

ORCID® identifiers
A. Mudadi
https://orcid.org/0009-0004-2104-865X

H.H. Lotriet
https://orcid.org/0000-0002-0353-5073

DOI
http://dx.doi.org//10.7166/34-1-2759

ABSTRACT

DevOps is a software development philosophy that involves the working together of development, operations, and other software development departments in order to accomplish common business objectives. However, very little has been done to establish the organisational impact of DevOps on information technology organisations; so an understanding of this phenomenon remains incomplete among practitioners and academic researchers. In this paper a framework for an appropriate DevOps restructuring approach is presented, based on the literature and on a single case study of a large multinational company with its headquarters in South Africa. The paper contributes to a better understanding of the organisational implications of the implementation of DevOps.

OPSOMMING

DevOps is 'n sagteware-ontwikkelingsfilosofie wat die samewerking van ontwikkelings-, bedryfs- en ander sagtewareontwikkelings-departemente behels om algemene besigheidsdoelwitte te bereik. Baie min is egter gedoen om die organisatoriese impak van DevOps op inligtingstegnologie-organisasies vas te stel; so 'n begrip van hierdie verskynsel bly onvolledig onder praktisyns en akademiese navorsers. In hierdie artikel word 'n raamwerk vir 'n toepaslike DevOps-herstruktureringsbenadering aangebied, gebaseer op die literatuur en op 'n enkele gevallestudie van 'n groot multinasionale maatskappy met sy hoofkwartier in Suid-Afrika. Die artikel dra by tot 'n beter begrip van die organisatoriese implikasies van die implementering van DevOps.

1. INTRODUCTION

DevOps is derived from the merging of two terms: development and operations. It is a practice in which different software development stakeholders work together with the main objective of continually delivering high-quality software to customers, using the most efficient systems [33]. Software of uncompromising quality and with high and uninterrupted speed facilitates rapid feedback loops that allow continuous product improvement and enable an organisation to gain a competitive edge in the market [47]. However, there is a pressing need in both the practitioner and the academic research communities to comprehend fully the restructuring process that an organisation needs to go through to implement DevOps successfully [26].

This paper contributes to meeting this need for knowledge by collating the existing literature on restructuring organisations during DevOps implementation, and combining this with a single case study of the restructuring processes during DevOps implementation in a large multinational company headquartered in South Africa. The paper thus provides a framework for DevOps restructuring that is appropriate for information technology organisations and that could inform decision-making during the process of migration to DevOps.

The rest of the paper is structured as follows: (1) A literature review section is presented; (2) the research methodology is discussed; (3) the findings of the study and a discussion of these are presented; and (4) some conclusions are drawn.

2. LITERATURE REVIEW

2.1. Restructuring required for DevOps implementation

2.1.1. Culture

When implementing DevOps, development and operations staff need to collaborate closely on a daily basis until their roles converge. This means that the development staff end up being able to do operations such as software deployment and system stabilisation in production, while the operations team become skilled in software development and associated activities [16]. Job rotation for developers and operators is an effective way of achieving a cross-functional team that can perform tasks for different departments [13]. An organisation needs to support this teamwork by recognising both operations and development departments as one department by giving them the same incentives and goals [20]. It is important that all team members have the freedom to make decisions and are empowered to share their thoughts and ideas without fear of being victimised [12], [45].

2.1.2. Automation

When implementing DevOps, the entire software system should be automated, including the deployment pipeline, testing, monitoring, and measurement [15]. Automation aims to minimise human involvement, which is slow and error-prone [18].

Automated deployment pipeline

An automated deployment pipeline is used for the fast, continuous, and efficient movement of high-quality software from the developers' computers to cloud-based test or production environments [30]. An organisation can deploy software to production many times a day once its deployment pipeline is implemented [17]. The deployment pipeline can be achieved by using tools such as the Microsoft Azure DevOps tool [23] or Jenkins [34], depending on the organisational requirements [38]. Decisions related to the implementation of a deployment pipeline need to consider the functionality of an organisation's current system [12].

Automated system monitoring

Automated system monitoring is a post-software deployment activity [37]. It is done by placing monitoring tools in environments such as production and testing to observe and record the day-to-day functionality of applications [31] and to ensure that software is delivered to the customer within agreed times and with the expected functionality [17]. If abnormal incidents occur in the system, automatic notifications are sent to the DevOps teams to take the necessary action [48]. Monitoring enables a DevOps team to get feedback about a software system's performance, which supports informed decision-making about product improvements [44], [1]. System monitoring helps to understand a software application's resource utilisation demands, and so makes it possible to implement the appropriate random access memory (RAM) and computer processing units (CPU) that are required for servers to maintain application stability [22]. All DevOps team members should be responsible for monitoring, as this would promote a sense of shared responsibility and teamwork that improves agility [28]. Historical monitoring logs can be analysed to enable developers to predict the chances of the success of upcoming software releases [5]. Choosing and setting up suitable monitoring tools often requires expert knowledge to avoid the risk of missing important functionalities [3].

Automated system measurement

Measurement is a process of getting system data that shows how the system functions in real time using metrics [32]. For DevOps process metrics, technology metrics, service metrics [7], and maturity models are used. These enable an assessment of the extent of DevOps transformation in an organisation and decision-making about areas that need improvement [26].

System measuring should focus on carefully selected key business areas and not on people, because human beings are able to manipulate activities in relation to measurement objectives [44].

2.2. Difficulties encountered during DevOps restructuring

2.2.1. Cultural challenges

The alliance between the development and operations teams might be jeopardised by unwillingness on either side to do extra work, especially if DevOps is perceived as a way to overuse members or to make them less competent [43]. In some organisations, developers or operators might already feel overworked and that they cannot afford to take on an extra responsibility [4]. In other situations, operations personnel could feel that development work is not part of their contract agreement [21]. Developers might dislike doing an operator's job, since it is not their expertise; and so they might not support the decision to collaborate [16]. Some senior employees might resist any form of change because they prefer their old and trusted approaches [23]. Building a DevOps team might not be possible in some organisations where laws and contract obligations do not allow developers to access production environments or operations staff to access development environments; so it is important to review these impediments first [36]. In some organisations where teams are located far away from each other, communication is mainly electronic, and synchronous collaboration might be difficult, and might impact negatively on efficiency and productivity [44]. To minimise this problem, teams should have frequent face-to-face meetings to enable team members to discuss work-related issues and to become familiar with one other [14]. The integration of development and operations teams is complicated, the process might take years to achieve [39].

A high level of dedication is required from each team member to build a DevOps culture successfully [17]. Having both developers and operators reporting to one manager might reduce work-related conflicts [20]. [29] noted that to mitigate challenges that result from development and operations teamwork it is important to communicate with employees to encourage a positive mindset and to invest in employee skills improvement training courses. [39] argues that a good management team is essential for an organisation to implement DevOps successfully, especially when a cultural transformation is required. [23] concluded that, for development and operations cooperation to reach its full potential, it is important that an organisation's top management recognise and support it.

2.2.2. Automation challenges

Automation is the use of tools or technology to accomplish tasks [29], Automation might, however, result in numerous complex difficulties in implementing DevOps.

Automated deployment pipeline challenges

Resistance to change: Managers who resist change can be big impediments to technological transformation [29]. Customers might not like the idea of having software delivered to them daily, and therefore an agreement must be in place before the deployment pipeline is functional [42]. Some customers complain that they do not see the new features that are claimed to be deployed every day using the deployment pipeline; therefore, an effective communication strategy should be prioritised to avoid misunderstandings [9]. In some organisations, production deployment cannot be done until all stakeholders have approved it, and this can have negative effects on the frequency and speed of production deployment [27].

Choosing the right tools: Choosing tools for the deployment pipeline is complex, and has a material impact on the success of DevOps adoption [4].

Continuous integration: Continuous integration in the deployment pipeline can become time-consuming and difficult when the code changes from different developers fail to work together and require a team effort to be corrected [24].

Resistance to use microservices: Although organisations are encouraged to change their software architecture from monolithic to microservices to enable deployment pipeline functionality, many organisations would resist such a change of architecture without proof of the benefits [43]. Some organisations prefer to develop new systems using microservices and to ensure that legacy systems remain operational in their existing state. However, developers often complain that the maintenance of legacy

systems distracts them from implementing new systems effectively [21]. Employees who are not familiar with microservices often find their features difficult to use. Thus there is need to simplify microservices adoption by using the fewest and most viable requirements [26]. Different components of software in microservices might behave differently, creating difficulties in ensuring that the application functions as expected across the entire system [8].

Test environment deployment: It is important to replicate the production environment in a test environment to ensure smooth deployments. However, the process of ensuring that production and test servers have the same configurations and features can be difficult because of the high costs involved and the strict production access controls and restrictions that some organisations impose [42]. Manual quality checks are often still used to determine software's readiness to be deployed to test or production servers. This is because, in most cases, efficient automatic rollback is not available if software with a bug is deployed [42].

Maintenance: For a deployment pipeline to operate with the required degree of efficiency and effectiveness, it needs to be maintained by employees with a variety of related skills and experience. This is often not the case, as many organisations have a shortage of required expertise [41], [30].

Different production environments: In organisations with multiple production environments, implementing an automated deployment pipeline is often complicated because of the different configurations and access requirements of these environments [43].

Software bugs: Having a fully functional and automated deployment pipeline does not guarantee the fast deployment of software features to production. In some instances, deployment can be delayed because of bugs caused by software integration activities [25].

Compliance requirements: In some cases, having an automated deployment pipeline that is operational can be a violation of governmental software regulation policies. So it is necessary to know the compliance requirements before implementation [42]. [30] added that a fully functional deployment pipeline can be highly problematic because it involves a significant change to the way software is managed. If software improvements include making changes to the database, implementing a deployment pipeline becomes extremely difficult.

Security: [46] warns that many organisations might implement a deployment pipeline that is susceptible to security attacks by hackers and malware because of a limited understanding of security configurations. Thus there is need to consider using virtual machines where possible, and consulting security experts.

Automated testing challenges: Automated testing does not guarantee flawless software because new errors caused by test cases can remain undetected. Therefore, it is necessary to use a variety of testing methods to mitigate this [6]. If an organisation has multiple environments that are not compatible with one another, automated testing becomes less effective [36]. Some organisations view automated testing as a nonviable option, as it involves high costs and low returns [42]. Some organisations end up with deployment pipelines that do not have performance testing techniques because they are too expensive and time consuming to establish, while other organisations have inadequate test coverage of their systems, thus allowing software flaws to find their way into production [3]. Testing results are sometimes not clear indicators of the causes of test success or failure, leaving developers confused about further testing steps [24].

Automated system monitoring challenges: It is hard to determine the part of the application that needs monitoring, as most monitoring activities are initiated after a software failure in a certain area of the application [30]. In some scenarios, where an organisation does huge daily software releases, analysing monitoring logs to identify problems might be labour-intensive and time-consuming, and it would need to be done with a sense of urgency [32]. The size of monitoring logs can grow rapidly and use up available storage space and memory, resulting in a need to upgrade the existing servers [42]. Unfortunately, there are still no monitoring tools that can predict application defects before they cause downtime [31].

Automated system measurement challenges: There is no specific way to identify the units of a system that needs to be measured, nor is there a standard way to measure whether an organisation has successfully implemented DevOps [29]. System measurement information that supports metrics selection is still very limited. As a result, many organisations use metrics that are based primarily on availability rather than

effectiveness [35]. Most of the tools that are used to measure system performance are not compatible with the tools used for the deployment pipeline, and so they remain unused [2]. Organisations also need to consider the impact of metrics on employee morale [26].

3. RESEARCH METHODOLOGY

3.1. Research philosophy

This research study assumed a qualitative epistemological stance, as its findings would be a result of field research based on the contributions from the setting in which the study took place [10]. The study employed both subjectivism and constructionism, as the researcher engaged with the participants to get their perceptions of how they changed their organisation for DevOps, and to learn how they understood DevOps, based on how they implemented it [10].

3.2. Selected research method

For this project a case study was selected as being the most appropriate method. A case study is a thorough investigation of a fact using real-life scenarios to establish a detailed relationship between the phenomenon and its surroundings [50]. Case study research explores a phenomenon without changing its natural setup in order to achieve a deeper and more accurate comprehension of the events, based on the interpretations and meanings given by the participants [11].

A case study approach is suitable when the aim of the study is to answer 'why' and 'how' questions, and when there is need for a richness of description in the research questions [38]. According to [49], the case study approach can be used when there are no clear boundaries between the phenomenon and the context. A suitable data collection plan must be achieved with the guidance of a case study protocol. A single case study is when a researcher investigates a phenomenon in a single unit of analysis [49]. To answer the research questions, a single case study was done in a large multinational organisation that practices DevOps.

3.3. Selected case study

The multinational company selected for the case study offers banking services in Africa and internationally, and is one of the largest financial services providers in Africa. Information technology (IT) is the backbone of their business, as they use software technology to provide services to their customers. The organisation ranks IT spending as one of their biggest expenses, which demonstrates their commitment to ensuring that their digital transformation meets international standards and uses the best technology. The employees who participated in this research are actively involved in the DevOps transformation of the company, and are based in South Africa. They manage different departments as key decision-makers who direct both the development and the operations of the organisation's IT infrastructure.

3.4. Data collection

Online interviews were conducted with four senior employees of the organisation who are key decision-makers in their departments, in order to understand the approach to organisational restructuring during DevOps implementation. The interviews were guided by these questions:

1. How did you restructure your organisation during DevOps implementation?
2. Are there any other changes related to DevOps transformation that occurred?
3. What challenges did you encounter during DevOps transformation?
4. What can you say are the possible solutions to the challenges?

3.5. Data analysis

The researcher first listed all key points raised during the interviews and grouped them per research question. NVivo qualitative data analysis software was used to code these grouped key points. Finally, the findings from the analysis were classified according to DevOps restructuring approaches, the challenges, related solutions, and the effectiveness of DevOps tools.

4. FINDINGS AND DISCUSSION

4.1. Restructuring approaches and changes during DevOps implementation

The respondents mentioned various approaches that could be followed to achieve DevOps adoption and, associated with these, indicated the changes that took place during the implementation. The aim of these were to minimise negative effects while maximising positive outcomes for the business. Each of the suggested approaches and changes is briefly discussed.

Cross-functional teams

"*You build it, you run it*" (Respondent 4). The interviewee added that teaming and proper tooling are an important part of DevOps transformation. Both developers and operators need to collaborate to accelerate software development and deployment. Respondent 1 also emphasised the need to have teams that are cross-functional to help improve software delivery. This corresponds with arguments in the literature on the need to have cross-functional teams when implementing DevOps [16].

Understanding your current business and its vision and mission

"*A good understanding of the goals and aspirations of the future [is] a critical factor when wanting to implement DevOps*" (Respondent 2). According to respondent 1 it is imperative that, before any DevOps changes can be made, the organisation should understand its own business and short- and long-term objectives. Respondent 4 added that adopting DevOps should be justified by the desired future position of the organisation. This links to the literature that emphasises the importance of an organisation understanding its current system [12].

Assessment of current technology usage and culture setup

"*[The] culture of the organisation is important to understand in terms of the shared values and behaviours of the employees*" (Respondent 3). Respondent 2 commented that an organisation must take note of the tools in current use and their effectiveness and efficiency in delivering the desired outcomes. Respondent 1 added: "*DevOps is about sharing, be it tools, responsibility or work, so a good understanding of the current culture will enable the company to focus on areas of improvement during restructuring*". Respondent 4 suggested that working as a team is important to achieve goals. The literature emphasises continuous technology assessment to ensure high quality [44].

Obtain information about the skills structure of the organisation

"*Sometimes you have to work with the human resources department to identify skills gap[s] and find ways to solve the problem*" (Respondent 2). Respondent 1 indicated that it is important to know the existing types of skills and experience that an organisation must take advantage of during the gradual transformation to DevOps, while simultaneously reskilling and sourcing external skills. Respondents 1, 2, and 3 emphasised that, as an organisation implements DevOps, the need for multi-skilled employees who can work in different departments – whether development, testing, or operations – becomes a fundamental requirement. The literature does not explicitly state the need to involve the human resources department to source more skills, but the need to hire employees is pointed out [40].

Do a DevOps impact assessment to find out how it will add value to the organisation

"*When you implement DevOps, you want to be confident that it is the right thing to do*" (Respondent 1). Respondents 1 and 2 pointed out that an organisation does not want to feel that DevOps was a wrong choice after implementation. Therefore, it is important to know the market in which the business operates, especially in relation to customer behaviour and demands. A conviction that the product on offer needs to be continuously improved and rapidly delivered is an indication that DevOps implementation might be appropriate. This point was not stated as clearly in the literature as it was by the respondents.

Be knowledgeable about the state of your business's operating model

"*Most businesses have legacy systems that are characterised by monolithic architecture and have on-premise data centres*" (Respondent 1). He added: "*You have to let go a lot of legacy things and embrace experimentation and improvement*". Respondent 1 also explained that the legacy systems are normally known as traditional operating models. The focus of these models is to maintain the stability and efficiency of the system while ensuring that each employee is assigned to a specific responsibility. Departments are siloed, with very limited collaboration in the organisation or with customers.

Respondents 1, 2, and 3 added that, when implementing DevOps, an organisation normally has to transform from the traditional model to a multi-model, and finally to a platform model. A multi-model is a gradual shift from the traditional model when an organisation begins to implement some DevOps practices, while a platform model is when the organisation has fully implemented all DevOps practices without any legacy systems still in operation.

The literature only indicates the need to shift to micro-services and the cloud for DevOps [8]. The respondents therefore provided more detail on organisational change during DevOps implementation.

Market the DevOps idea in the organisation to change the employees' mindset

"*There is need to market the idea of DevOps to the organisation to convince the employees that it is the right thing to do*" (Respondent 2). Respondents 1 and 3 emphasised the importance of continuous sharing of information with employees about DevOps practices such as automation, close collaboration, multi-skilling, and the associated benefits. If employees are to collaborate, they need to be convinced about the benefits of DevOps to the organisation and to individual employees. This supports the view in the literature that influencing role players' mindsets to participate positively in change is a critical first step [29].

Design DevOps maturity model based on DevOps best practices

"*DevOps maturity assessment has got six functions; we look at continuous planning, continuous development, continuous testing, continuous deployment, continuous monitoring, and continuous logging*" (Respondent 1). He added that it is important that, before or during DevOps implementation, an organisation develop a DevOps maturity model to measure progress at every stage, and so improve. According to respondents 1, 2, and 4, the DevOps maturity model consists of DevOps expectations at each stage of the DevOps implementation. Since adopting DevOps is a process, different stages of the process should have checklists of the activities to be achieved with DevOps continuous practices such as automation, integration, deployment, delivery, monitoring, best use of toolsets, close collaboration, and learning and improvement. In addition to the checklist, the organisation needs to devise a strategy of how each step is to be achieved. Setting up different expectations at each stage of implementation would act as a guideline to motivate and push the team to achieve the desired results. This confirms the view in the literature that DevOps maturity models are important to measure DevOps implementation progress [14].

Identify the pilot project with which to start testing DevOps practices

"*It is important that, when you implement DevOps for the first time, you do not start it on a big project; rather start it on a small project that is easy to manage*" (Respondent 1). Respondents 3 and 4 added that it is necessary that different departments start collaborating on the pilot project, sharing ideas on how best to automate and increase the efficiency of the pilot project. This could be used as a first experiment to change the technology and the culture. It is imperative, therefore, that successes and failures be noted as lessons learnt for further projects. The literature does not mention identifying a pilot project.

Experimentation and learning fast

Respondent 2 commented: "*People should be willing to learn, experiment, and fail fast*". Respondent 4 added that experimenting during development is important for creativity. Experimenting is one of the requirements for a DevOps culture [40].

Use DevOps metrics to measure your progress and keep on improving (a change related to DevOps transformation)

Respondent 2 emphasised: *"It is important that you measure the improvements of DevOps to understand your successes and failures*". Respondent 1 added that it is important to go beyond the quality assurance (QA) department's scope of work by measuring the system's performance in real time. Respondents 2, 3, and 4 commented that the ability to measure the functionality of the software in production provides insights into the usefulness of the software, and enhances the decision-making process for fixes and improvements. The literature also highlights the need to use tools that measure software performance and provide continuous feedback [47].

Ensure system security is part of every stage of the DevOps implementation (a change related to DevOps transformation)

"*Everything that we do must have a security application*" (Respondent 1). Respondent 2 commented that there is need to ensure that the system is highly secured during DevOps transformation to protect it against unauthorised users and hackers. Every time the system is improved, the security needs to be reviewed and adjusted accordingly to ensure that it remains relevant and effective.[46] notes that security must be part of every DevOps process.

Use of dashboards (a change related to DevOps transformation)

"*Teams could go to DevOps dashboards and see how they are performing against the targets that they have set themselves*" was the point raised by respondent 4 to explain the use of dashboards. According to the interviewee, the dashboards helped to enforce transparency in the organisation, as employees could see the loads that had been successfully processed and those that had failed. The need for dashboards is also noted in the literature [19].

Stage 1 *Gather information*	*Stage 2* *Analyze*	*Stage 3* *Implement in phases*	*Stage 4* *Measure to improve*
Understand your business, its vision and mission. Gather information about your organisation's technology, culture and skills.	Do a DevOps impact assessment to determine viability. Analyze your business operating model. Design DevOps maturity model. Market the benefits of DevOps in your organisation.	Identify pilot project to start DevOps with. Use your DevOps maturity model to identify technology and culture changes. Get lessons from pilot project and implement DevOps on a bigger project. Gradually implement on other projects.	Use DevOps metrics to measure successes and failures. Keep on improving. Keep on learning and experimenting.

Figure 1: DevOps restructuring summary suggested by respondents

4.2. Restructuring challenges and solutions

A further theme that was explored in the case study related to restructuring challenges and the potential solutions to them, as suggested by the participants. Table 1 summarises these challenges and the participants' suggested solutions.

Table 1: Summary of restructuring challenges and proposed solutions

Restructuring challenge	Related suggested solution
Manual intervention	▪ Automate the entire workflow to minimise human involvement
Legacy change management processes	▪ Ensure that there is accountability and a realistic turnaround time for every approval process ▪ Automate some of the approval processes
High DevOps success rate concentrated on pilot projects while limited across all projects	▪ Set realistic, achievable, and timeous goals, and track them and share findings ▪ Subscribe to uniform and best standards of development and deployment to ensure stability and repeatability ▪ Create a DevOps manifesto based on past failures and successes, and learn from it to improve ▪ Ensure that there are fast feedback loops in the deployment pipeline to fail fast and recover fast
Unwillingness to learn new skills and processes	▪ Effective communication to convince employees about the benefits to them of the changes ▪ Ask human resources department to help fill the skills gap
Employees located far away from each other	▪ Use online video conferences ▪ Organise team-building sessions to meet and talk
Indecision when it comes to which part of the business to outsource during restructuring	▪ Communicate and thoroughly consult with all business stakeholders to help make a good decision ▪
Automating waste, creating automation before fixing bottlenecks and critical infrastructure points	▪ The first step is to identify bottlenecks and points of weakness, and then to use automation where possible to solve them
Changing management employees	▪ Embrace new ideas from new employees, but ensure that the ideas are meant to improve the workflow in line with the vision of the organisation
Outdated testing practices	▪ Make testing and quality assurance everybody's responsibility; let developers also test and testers also develop
Not taking IT governance seriously	▪ Establish governance to set required standards that control IT processes, and make everyone accountable
Lack of shared ownership and accountability because of silos	▪ Continuously assess your operational processes such as access management, incident management, request management, and problem management so that you can improve their agility and transparency
Use of different technologies and different configurations in the same environment	▪ Standardise infrastructure technology to be identical and to subscribe to the best practices

Continue next page

Table 1: Summary of restructuring challenges and proposed solutions (cont.)

Restructuring challenge	Related suggested solution
Achieving a high-maturity DevOps is a process, not an event	▪ Be patient, as a high-velocity organisation cannot be achieved overnight, and stick to basics ▪ Create an environment that promotes continuous learning and improvement
Lack of operational maturity	▪ Increase collaboration between different departments ▪ Promote cross-skilling, and market the benefits of doing that to the employees
Silo mentality	▪ Create cross-functional teams ▪ Encourage the required behaviour by giving incentives and rewards
Lack of executive management support	▪ Take initiatives from the bottom to implement and ensure that everybody understands DevOps ▪ Use metrics to measure successes, and then use positive outcomes to justify to the executive the need for DevOps and so obtain their support

5. LIMITATIONS AND FUTURE RESEARCH

Although the unit of analysis operates in many countries, the findings are based on a case study conducted in the South African context, and thus they might not be representative of other contexts. The research was only focused on the impact that DevOps has on IT organisations; however, it is also important for research to be conducted to understand the ways in which DevOps affect the customers of these organisations.

Based on these limitations, future research into the organisational aspects of DevOps could focus on diverse contexts with different market environments. The scope of the research would need to be expanded to include the impact of DevOps adoption on customers who are affected.

6. CONCLUSIONS

This analysis of the impact of DevOps implementation on IT organisations has made it clear that the main changes that take place in organisations during DevOps implementation relate to their culture and technology. The cultural aspect, which involves changing employees' mindsets, is the first critical step in DevOps transformation; otherwise the changes in technology will not bring the expected success.

Both the literature review and the field research have shown that the impact of DevOps on IT organisations occurs in three forms. The first is workflow optimisation, which involves changes in the organisation to ensure that DevOps practices lead to increased productivity. The second is fast feedback loops. The entire organisation restructures itself to enable effective communication to be a priority in every action and in every process. The main benefits are the ability to fail and recover quickly and to meet customer requirements. The third form is the establishment of a culture of continuous learning and experimentation. Key to leadership in DevOps is having employees who have a passion for, and are willing to keep on learning about, new technology and to experiment with existing technology. This results in technological innovation, which is one of the primary drivers of organisational growth and increased market share.

The focus of the paper was on the influence that DevOps implementation has on IT organisations. The literature review and the field research have contributed to understanding the key factors that are part of the organisational evolution during DevOps implementation.

This research experience showed that DevOps brings about important technological and cultural changes that could help IT organisations to become innovative and competitive. The future of digital transformation is indeed DevOps; and the more that IT organisations embrace this software development methodology, the more the future of technology will be exciting and life-changing.

REFERENCES

[1] **Akshaya, H., Vidya, J., & Veena, K.** 2015. A basic introduction to DevOps tools. *International Journal of Computer Science & Information Technologies,* 6(3), pp 2349-2353.

[2] **Bezemer, C., Eismann, C., Ferme, V., Grohmann, J., Heinrich, R., Jamshidi, P., & Willnecker, F.** 2019. How is performance addressed in DevOps? In *Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering*, pp 45-50.

[3] **Brunnert, A., Van Hoorn, A., Willnecker, F., Danciu, A., Hasselbring, W., Heger, C., & Koziolek, A.** 2015. Performance-oriented DevOps: A research agenda. *arXiv preprint arXiv: 1508.04752*, pp 1-46.

[4] **Bucena, I., & Kirikova, M.** 2017. Simplifying the DevOps adoption process. In B. Johansson (ed), *BIR Workshops*, pp 1-15. Aachen: CEUR Workshop Proceedings.

[5] **Capizzi, A., Distefano, S., Araújo, L., Mazzara, M., Ahmad, M., & Bobrov, E.** 2019. Anomaly detection in DevOps toolchain. In J. Bruel, M. Mazzara & B. Meuer (eds), *International Workshop on Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*, pp 37-51. Cham: Springer Nature.

[6] **Caprarelli, A., Nitto, E., & Tamburri, D.** 2019. Fallacies and pitfalls on the road to devops: A longitudinal industrial study. In J. Bruel, M. Mazzara & B. Meuer (eds), *International Workshop on Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*, pp 200-210. Cham: Springer Nature.

[7] **Cartlidge, A., Hanna, A., Rudd, C., Macfarlane, I., Windebank, J., & Rance, S.** 2007. *An introductory overview of ITIL V3*. Bracknell: The UK Chapter of the itSMF.

[8] **Chen, L.** 2018. Microservices: Architecting for continuous delivery and DevOps. In *2018 IEEE International Conference on Software Architecture (ICSA)*, pp 39-397.

[9] **Claps, G., Svensson, R., & Aurum, A.** 2015. On the journey to continuous deployment: Technical and social challenges along the way. *Information and Software Technology,* 57, pp 21-31.

[10] **Crotty, M.** 1998. *Foundations of social research: Meaning and perspective in the research process*. London: SAGE.

[11] **Crowe, S., Cresswell, K., Robertson, A., Huby, G., Avery, A., & Sheikh, A.** 2011. The case study approach. *BMC Medical Research Methodology,* 11(1), pp 1-9.

[12] **Crowley, C., McQuillan, L., & O'Brien, C.** 2018. Understanding DevOps: Exploring the origins, composition, merits, and perils of a DevOps capability. In *Proceedings of the 4th International Conference on Production Economics and Project Evaluation*, pp 29-37.

[13] **De França, B., Jeronimo, H., & Travassos, G.** 2016. Characterizing DevOps by hearing multiple voices. In *Proceedings of the 30th Brazilian Symposium on Software Engineering*, pp 53-62.

[14] **Díaz, J., Perez, J., Yague, A., Villegas, A., & De Antona,** A. 2019. DevOps in practice: A preliminary analysis of two multinational companies. In *International Conference on Product-Focused Software Process Improvement*, pp 1-8.

[15] **Diel, E., Marczak, S., & Cruzes, D.** 2016. Communication challenges and strategies in distributed DevOps. In *IEEE 11th International Conference on Global Software Engineering*, pp 24-28.

[16] **Erich, F., Amrit, C., & Daneva, M.** 2017. A qualitative study of DevOps usage in practice. *Journal of Software: Evolution and Process,* 29(6), pp 1-47.

[17] **Gall, M., & Pigni, F.** 2021. Taking DevOps mainstream: A critical review and conceptual framework. *European Journal of Information Systems,* 31(5), pp 1-20.

[18] **Gill, A., Loumish, A., Riyat, I., & Han, S.** 2018. DevOps for information management systems. *VINE Journal of Information and Knowledge Management Systems,* 48(1), pp 122-139.

[19] **Humble, J., & Farley, D.** 2010. *Continuous delivery: Reliable software releases through build, test and deployment automation.* Boston, MA: Pearson Education.

[20] **Hüttermann, M.** 2012. *DevOps for developers,* 1st ed. New York, NY: Apress.

[21] **Jones, S., Noppen, J., & Lettice, F.** 2016. Management challenges for DevOps adoption within UK SMEs. In *Proceedings of the 2nd International Workshop on Quality-aware DevOps*, pp 7-11.

[22] **Karamitsos, I., Albarhami, S., & Apostolopoulos, C.** 2020. Applying DevOps practices of continuous automation for machine learning. *Information*, 11(7), pp 1-15.

[23] **Khan, M., Shaikh, A., & Farhan, W.** 2020. Fast delivery, continuously build, testing and deployment with DevOps pipeline techniques on Cloud. *Indian Journal of Science and Technology,* 13(5), pp 552-575.

[24] **Laukkanen, E., Itkonen, J., & Lassenius, C.** 2017. Problems, causes and solutions when adopting continuous delivery — A systematic literature review. *Information and Software Technology,* 82, pp 55-79.

[25] **Lehtonen, T., Suonsyrja, S., Kilamo, T., & Mikkonen, T.** 2015. Defining metrics for continuous delivery and deployment pipeline. In J. Nummenmaa, O. Sievi-Korte & E. Mäkinen (eds), Proceedings of the 14th Symposium on Programming Languages and Software Tools (SPLST'15), pp 16-30. Aachen: CEUR Workshop Proceedings.

[26] **Leite, L., Rocha, C., Kon, F., Milojicic, D., & Meirelles, P.** 2019. A survey of DevOps concepts and challenges. *ACM Computing Surveys,* 52(6), pp 1-35.

[27] **López-Fernández, D., Diaz, J., Garcia-Martin, J., Pérez, J., & Gonzalez-Prieto, A.** 2021. DevOps team structures: Characterization and implications. *IEEE Transactions on Software Engineering,* 14(8), pp 1-18.

[28] **Luz, W., Pinto, G., & Bonifácio, R**. 2018. Building a collaborative culture: A grounded theory of well succeeded DevOps adoption in practice. In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pp 1-10.

[29] **Luz, W., Pinto, G., & Bonifácio, R**. 2019. Adopting DevOps in the real world: A theory, a model, and a case study. *Journal of Systems and Software,* 157, pp 1-35.

[30] **Lwakatare, L., Kilamo, T., Karvonen, T., Sauvola, T., Heikkilä, V., Itkonen, J., & Lassenius, C.** 2019. DevOps in practice: A multiple case study of five companies. *Information and Software Technology,* 114, pp 1-18.

[31] **Lwakatare, L., Kuvaja, P., & Oivo, M.** 2016. An exploratory study of DevOps extending the dimensions of DevOps with practices. In L. Lavazza, M. Kajko-Mattson, K.M. Ravi, R. Koci & S. Clyde (eds), ICSEA 2016 The Eleventh International Conference on Software Engineering Advances, pp 91-341. Rome: IARIA.

[32] **Lwakatare, L., Kuvaja, P., & Oivo, M.** 2015. Dimensions of DevOps. In *International Conference on Agile Software Development*, pp 212-217.

[33] **Mangot, D**. 2016. *Mastering DevOps*. Retrieved June 4, 2020 from https://learning.oreilly.com/videos/masteringdevops/ 9781786468048/9781786468048-video1_3

[34] **Mohammad, S.** 2016. Continuous integration and automation. *International Journal of Creative Research Thoughts,* 4(3), pp 938-944.

[35] **Ravichandran, A., Taylor, K., & Waterhouse, P.** 2016. *DevOps for digital leaders: Reignite business with a modern DevOps-enabled software factory*. New York: CA Press.

[36] **Riungu-Kalliosaari, L., Mäkinen, S., Lwakatare, L., & Männistö, T.** 2016. DevOps adoption benefits and challenges in practice: A case study. In *International Conference on Product-Focused Software Process Improvement*, pp 590-597.

[37] **Rowse, M., & Cohen, J**. 2021. A survey of DevOps in the South African software context. In *Proceedings of the 54th Hawaii International Conference on System Sciences*, pp 6785-6794.

[38] **Rowley, J.** 2002. Using case studies in research. *Management Research News*, 25(1), pp 16-27.

[39] **Rütz, M.** 2019. DevOps: A systematic literature review. *IT Management Seminar Paper Summer Term*, pp 1-11.

[40] **Sánchez-Gordón, M., & Colomo-Palacios, R**. 2018. Characterizing DevOps culture: A systematic literature review. In *International Conference on Software Process Improvement and Capability Determination*, pp 1-13.

[41] **Senapathi, M., Buchan, J., & Osman, H**. 2018. DevOps capabilities, practices, and challenges: Insights from a case study. In *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018*, pp 1-11.

[42] **Shahin, M., & Babar, M.** 2020. On the role of software architecture in DevOps transformation: An industrial case study. In *Proceedings of the International Conference on Software and System Processes*, pp 175-184.

[43] **Smeds, J., Nybom, K., & Porres, I**. 2015. DevOps: a definition and perceived adoption impediments. In *International Conference on Agile Software Development*, pp 166-177.

[44] **Teixeira, D., Pereira, R., Henriques, T., Silva, M., & Faustino, J.** 2020. A systematic literature review on DevOps capabilities and areas. *International Journal of Human Capital and Information Technology Professionals,* 11(3), pp 1-22.

[45] **Van Belzen, M., DeKruiff, D., & Trienekens, J.** 2019. Success factors of collaboration in the context of DevOps. *International Conference Information System,* pp 26-30.

[46] **Wilde, N., Eddy, B., Patel, K., Cooper, N., Gamboa, V., Mishra, B., & Shah, K**. 2016. Security for Devops deployment processes: Defenses, risks, research directions. *International Journal of Software Engineering & Applications,* 7(6), pp 1-16.

[47] **Wurster, L., Colville, R., Haight, C., Tripathi, S., & Rastogi, A**. 2013. Emerging technology analysis: DevOps a culture shift not a technology. pp 1-12. Gartner.

[48] **Yarlagadda, R.** 2021. DevOps and its practices. *International Journal of Creative Research Thoughts,* 9(3), pp 2320-2882.

[49] **Yin, R.K.** 2018. Case study research and applications, 6th ed. Thousand Oaks, CA: SAGE Publications.

[50] **Zainal, Z**. 2007. Case study as a research method. *Jurnal Kemanusiaan*, 5(1), pp 2-6.