

## A Survey of DevOps in the South African Software Context

Morgan Rowse  
 University of the Witwatersrand  
[morgan.rowse@gmail.com](mailto:morgan.rowse@gmail.com)

Jason Cohen  
 University of the Witwatersrand  
[jason.cohen@wits.ac.za](mailto:jason.cohen@wits.ac.za)

### Abstract

*This study investigated DevOps practices and experiences in the South African software development context, along with associated perceptions of benefits and challenges. Survey data collected from a sample of 80 software development professionals showed that more frequent builds, earlier detection of bugs and reduced project lead times were the top three benefits, while getting DevOps capable members into a team, finding experienced professionals to support DevOps practice and changing deep-seated company culture to support DevOps were the top three challenges. DevOps practices are still emerging. Although 85% of respondents report continuous integration as a frequent practice, only 54% report using continuous deployment frequently. The biggest reported impacts of DevOps on software development culture were in making development team members aware of operational faults, responsible for quality assurance, and responsible for deployments. Realisation of benefits from DevOps depends largely on a culture change. Results are useful for guiding organisations considering DevOps adoption.*

### 1. Introduction

Software development and operations teams have traditionally operated with a segregation of responsibilities [1, 2, 3]. While these silos may have helped to reduce risks to production environments, they are known to cause long delays between code updates and deployment. Tensions frequently arise as developers try to meet customer demands by pushing new software releases while operations tries to maintain stability and reliability of software in production [4, 5]. Yet, business increasingly demands more frequent releases and cannot afford for delivery of software into production to be a time-consuming, stressful and error-prone process [6].

DevOps (Development & Operations) is an emerging paradigm for software development and operations [7, 8]. DevOps adopts lean process principles with a focus on eliminating unnecessary handovers,

waiting time, waste and rework in software development and deployment. DevOps is both a culture shift in software delivery [9], and a set of practices for continuous integration, delivery and deployment (CI/CD) supported by technology enablers [10]. By combining new ways of working along with improved communication and a high degree of automation, DevOps aims to solve the problems between development and operations [4]. DevOps offers the promise of improved software quality, faster software release cycles, more frequent deployment of new features with better reliability, and earlier delivery of business value [7, 11, 12].

However, the adoption of DevOps into software practice has not been without its problems. Adoption has been described as sluggish and often limited just to smaller teams and projects [13]. Problems with DevOps arise, in part, because it demands cross-functional competencies, hard and soft skills in the IT team [3, 14]. Organisations often confront difficulties integrating DevOps methodologies into their existing projects due to the complexity of prior deployment processes as well as a lack of technical deployment ability among their software developers [15]. DevOps has also been associated with structural and task-related changes, greater responsibilities, blurred lines of authority, threats to job security and complex performance metrics that can collectively lead to resistance from IT development and operations professionals [11]. DevOps efforts have also failed in some environments [16], or have been difficult to sustain and integrate into software development culture [17]. DevOps has also been implemented in different ways and to different extents across organisations [12]. There is lack of clarity on the appropriateness of DevOps practices and a lack of evidence on the benefits and challenges associated with building DevOps expertise and culture [18]. While a growing body of work aims to define and clarify DevOps concepts (e.g. [8]), there is limited empirical research into DevOps experiences in industry.

We were thus motivated to contribute to a better understanding of DevOps in industry. The study will explore DevOps practices and provide insight into practitioners' perspectives on their DevOps culture, its

benefits as well as any challenges to be expected during the integration of DevOps into software projects. Our research questions are: *RQ1*: What DevOps practices are most often applied? *RQ2*: How is DevOps influencing the culture of software practice? *RQ3*: What are the perceived benefits of DevOps for software practice? *RQ4*: What are the challenges to DevOps?

The context for our empirical work is the South African software industry. South Africa has a relatively small but highly promising software industry. Software spending is growing but the country still confronts large technical debt and a significant skills shortage [19]. Improvement in software delivery is viewed as essential to its ongoing digital transformation and global competitiveness [20].

## 2. Related work

Extant literature defines DevOps with an emphasis on DevOps culture and its reliance on a set of software engineering practices. Leite et al. [6] carried out a literature review of motivations for DevOps adoption and found that while technical issues regarding DevOps automation are more clearly documented in literature, there is not yet consensus on how to effectively implement DevOps culture. Unfortunately, relatively few studies have focused on illustrations of DevOps practice and culture in industry. Gupta et al. [21] carried out one of the few large scale studies to identify the attributes useful for evaluating the maturity of a DevOps implementation. They collected data from 300 respondents from 10 multi-national software development organisations in order to classify the key attributes of DevOps. They found source control and automation tools were important to realising continuous delivery principles. They also defined cohesive teams as an important category of DevOps attributes [21]. Luz et al. [22] interviewed DevOps practitioners from 15 companies across Brazil, Ireland, Portugal, Spain, and United States. Their work added to understanding of successful paths to DevOps adoption. They defined enablers and outcomes of DevOps as including automation, transparency and sharing, agility, resilience, continuous measurement and quality assurance. Most importantly, they found that a collaborative culture that removes silos between development and operations is the core concern for DevOps, and that it is not sufficient to achieve DevOps by focusing only on automation tools for building, deployment and infrastructure provisioning.

Prior research into the challenges of DevOps has mostly been limited to small qualitative case studies. For example, Lwakatare et al. [13] carried out a qualitative study of four companies in Finland to identify challenges to DevOps. They concluded that hardware

dependency, software incompatibility, limited visibility of customer environments, and lack of technology to automatically deploy new features into customer environments were among the top challenges. Riungu-Kalliosaari et al. [23] carried out four qualitative interviews in three case study organisations in Finland to identify benefits and challenges of DevOps. They found challenges largely surrounded culture and implementation of DevOps across heterogeneous environments, while benefits included more implemented features and frequent releases, better quality of releases from automation along with enhanced collaboration and communication between functions and well-being of the DevOps team. Amaradri and Nutalapati [24] show in the Swedish context that increased collaboration, fast releases and less stressful deployments are among the benefits of DevOps but DevOps teams often have problems with communication, entrenched differences, and picking the right tools. Jones et al. [25] carried out a small qualitative case study of a UK based SME. They found DevOps was not only a technical phenomenon but also a social and cultural one. They highlighted the management structure challenges to DevOps, along with unclear responsibilities between development and operations units. They also conclude that developers are often still faced with maintaining legacy systems, which interrupts their learning of new technologies required for DevOps. Kamuto and Langerman [26] carried out a small sample qualitative study in the South Africa software industry. They found the main factors hindering the adoption of DevOps were a lack of strategic direction from senior management, lack of DevOps education, concerns over job roles, resistance to change and a silo mentality. Senapathi et al. [10] present a case study of DevOps implementation in New Zealand. Interviews with six practitioners was used to conclude that DevOps benefits included an increase in deployment frequency as well as happier teams with improved communication and collaboration. They identified that both automation tools and changes to organizational culture and structure were important for delivery of DevOps benefits. Availability of technical skills, difficulty in setting up the tools for the deployment pipeline along with resistance to change and a lack of clear responsibilities were among the challenges associated with DevOps that needed to be overcome. Erich et al. [27] carried out exploratory interviews in 6 organisations to identify problems encountered in adopting DevOps. They similarly found a need for various technical and interpersonal skills as well as for collaboration. This need arises because the development and operations pipelines includes people from various disciplines contributing to the technical artifacts and components processed by the pipeline.

Concerns over DevOps included the increase in responsibilities, difference in work styles, and lack of evidence of DevOps effectiveness. Shropshire et al. [11] also found that IT professionals have concerns over the implications of DevOps. Through a survey of 271 technology workers, they found DevOps associated with increased perceptions of uncertainty over outcomes and changes required and that its implementation therefore has the potential to reduce morale. Hamunen [28] interviewed a small sample of IT professionals in Finland and cited lack of awareness and lack of management and team support for DevOps as particular challenges, along with difficulties adapting to the DevOps processes. A lack of skills in automation tools was also a frequently mentioned problem. Hemon et al. [3] carried out a case study within a single large company to describe how a transition from agile to DevOps relies on both hard and soft skills. They found technical hard skills were important at all levels of automation i.e. agile, through continuous integration and continuous delivery. However, soft skills and collaboration intensified with the level of automation. Khan and Shameem [29] identified 16 possible DevOps challenges. Based on data from a survey of 79 practitioners, they classified these into three high level categories reflecting people, business and change related challenges. Kuusinen et al. [30] carried out surveys and interviews in a Danish software service house on the challenges to reaching DevOps. Their data suggests most participants were not certain that continuous practices would mitigate problems in software development. Moreover, a transition to DevOps would require a major cultural change with additional challenges due to organisational structure and legacy software incompatibility.

In other related work, Wiedemann and colleagues have carried out a number of studies into the skills required for DevOps practitioners [14, 31, 32]. Using qualitative case studies and interviews, a number of DevOps competencies and skills were identified as important to software process improvements.

Taken together past research has given insight into potential benefits along with challenges associated with DevOps, and helped to highlight the relative roles of both culture and continuous engineering practices within DevOps. However, most work involves small sample interviews in a small number of organizational contexts and a comprehensive picture of DevOps practices and experiences has yet to emerge.

### 3. Research variables

From the literature, we identify DevOps practices, DevOps culture, DevOps benefits and DevOps

challenges as the most important factors relevant to our descriptive account of DevOps experiences in industry.

*DevOps Practices:* DevOps practices are focused on the deployment pipeline (refer Figure 1). First, source code, build scripts and tests are maintained in version control, then whenever there is a commit to this repository, a hook will automatically trigger a new build. This process will build the project and test it, and if the build and tests pass successfully then binaries will be automatically deployed to further environments e.g. quality assurance (QA). The DevOps teams approve changes and deploy to further staging environments or the production environment itself. If there is a breakdown during the building, testing and deploying process, the team will be notified of the failure [4]. The DevOps pipeline is summarised to include the stages of continuous integration, continuous testing, continuous delivery and deployment, and continuous monitoring [21]. Through these continuous engineering practices, DevOps supports the ‘release early, release often’ philosophy [7].

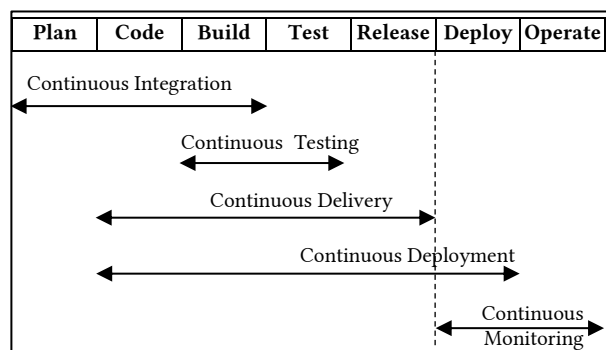


Figure 1. Deployment Pipeline (source: [21])

Continuous Integration is the practice of automated compilation, building and testing [33]. Continuous Testing is the activity where code is assessed against functionality, security, performance and acceptance tests by means of automatically conducted unit and feature tests [7]. Static analysis may also form part of the pipeline by enforcing code standards as well as to provide earlier detection of errors [34]. Continuous Delivery extends the process where successful software builds can be reliably released into the next staging environment following previous pipeline activities, although this may not necessarily be to end users [27, 35]. The DevOps teams will then decide whether and when to push these builds into production environments. Continuous Deployment differs from continuous delivery as it includes automatic deployment into production environments where end user will have access to new release features and their live data [7]. Continuous Deployment is thus tasked with fully automating production deployments while Continuous

Delivery is tasked with manual deployment to allow for more intervention on whether release-ready builds are pushed into production [36]. Continuous Monitoring is the automated process where cloud services are leveraged to test operational functionality of a deployed product at run time against business requirements [7].

*DevOps Culture:* DevOps is arguably more of a culture shift within IT than a process or tool shift [4, 37]. DevOps emphasizes a new culture in which development and deployment tasks are shared between developer and operation teams and where all team members accept responsibility for the quality of the final deliverable [7, 32]. DevOps sees operations engineers picking up development tasks and developers working in an operations environment [40]. DevOps requires communication and collaboration between development and operations [37]. For example in a DevOps culture, operations representatives should attend the meetings traditionally assigned to development and project management teams [4]. Similarly, developers should also concern themselves with operational faults and improve their knowledge of operations related errors to help prevent or reduce the likelihood of such faults in future revisions [4]. Thus developers share responsibility for quality assurance and operations stability. In a DevOps culture, the entire team takes responsibility for the quality of the delivered project rather than deferring onto other members. DevOps culture is also characterised by improved communication between development and operations units, sharing of knowledge around issues in development and production, and a constructive failure review process [4, 5, 7, 32].

*DevOps Benefits:* While some studies suggest that adopting DevOps can create uncertainty over responsibilities [11], other evidence suggests a DevOps culture can improve relationships between development and operations along with the well-being of developers [23]. In addition, DevOps benefits are considered to include improvements to deployment frequency, shorter lead time for changes, quicker time to recover and a reduction in failure rates [21]. A reduction in lead times also provides reduced times to user feedback, allowing for aspects such as user acceptance and satisfaction to play a large role in the development functions of a project, reducing possible rework [24]. DevOps also facilitates improvements to quality assurance, testing, developer productivity and a larger set of delivered features through intensified feedback cycles [23]. By enabling more frequent releases and automated deployments, DevOps can aid business to achieve competitive advantages through software [31].

*DevOps Challenges:* There is a general lack of DevOps awareness and maturity and a lack of experienced DevOps practitioners in industry [24, 28].

DevOps challenges can thus extend across the hard and soft dimensions of the concept. For some organisations, difficulty and unfamiliarity in using the automation tools required for DevOps may be a barrier [39]. Creating and replicating the complex pipeline environments required for a DevOps approach can also be a technical challenge [23, 28]. Legacy and currently running projects may prove challenging environments in which to integrate DevOps, especially where they present architectural incompatibility with DevOps practices [24]. Providing security to deployment pipelines and reducing threats introduced by the removal of segregation of duties may present additional difficulties for some [2].

For others, challenges may relate to implementing DevOps culture, which is often not well understood by software practitioners [21]. DevOps challenges typically arise from deep-seated company culture and difficulty adapting organisational processes to align with the DevOps approach [28, 38]. Implementing DevOps thus requires stakeholder engagement and involvement of IT leaders and management support [28, 41]. Getting all team members to take responsibility for product success requires changing mindsets [39]. Change management issues can arise [11, 38]. DevOps may also be associated with hidden costs. For example, a new DevOps workflow can initially be accompanied by inefficiencies and additional costs for DevOps implementation may be unanticipated [39].

## 4. Methods

To examine the four dimensions of the DevOps experience, a survey methodology was adopted to collect data from a sample of South African software development and IT operations professionals. The sample was a non-probability judgement sample that included selected professionals listed on LinkedIn, members of the community forum “ZA Tech” (zatech.co.za), and others from the authors’ network.

Tables 1 through 4 detail the items included in the survey instrument. All study items were developed from the literature. DevOps practices were examined through 5 items asking about the frequency with which continuous practices were applied to projects (from 1=never to 5=always). We focused on a core set of continuous practices [21] derived from the more comprehensive set of practices [8]. DevOps culture was reflected in 7 items. Respondents were asked to indicate their perceptions on 11 DevOps benefits and 15 DevOps challenges, all measured on an agreement scale (from 1=strongly disagree to 5=strongly agree). Face and content validity checks help ensure the suitability of the questionnaire. Content validity is concerned with whether the questionnaire adequately represents the

constructs under study. Face validity ensures the questions are clear and have meaning to respondents. We established face and content validity by firstly drawing on literature for development of the items and secondly by ensuring the survey was pre-tested by researchers and then pilot tested with three DevOps practitioners. The final questionnaire was distributed as an online survey to respondents via the Internet.

Table 1. DevOps Practices

DevOps Practices [7, 21, 33]	
Continuous Integration	How often do you use continuous integration tooling in your projects?
Continuous Testing	How often do you use continuous testing tooling in your projects?
Continuous Deployment	How often do you use continuous deployment tooling in your projects?
Continuous Monitoring	How often do you use continuous monitoring tooling in your projects?
Static Analysis	How often do you use static analysis tooling in your projects?

Table 2. DevOps Culture

DevOps Culture [4, 7, 27, 39]	
Developer Responsibility	DevOps is used to give our development team responsibility for deployment functions.
Quality Assurance	DevOps is used to give our development team responsibility for quality assurance functions.
Team Responsibility	DevOps is used to give our team members responsibility for deployment operations.
Communication	DevOps is used for greater communication between development and operational functions.
Feature ownership	DevOps is used for greater ownership of product features.
Operations attendance	DevOps is used so that operational representatives attend planning and development meetings.
Developer awareness	DevOps is used to make members of the development team aware of operational faults.

Table 3. DevOps Benefits

DevOps Benefits [12, 21, 23, 24, 31]	
Lead times	DevOps allows for reduced project lead times.
Deployment automation	DevOps allows changes to be automatically deployed without human intervention.
Development pace	DevOps enables a faster development process.
Build frequency	DevOps enables more frequent builds.
Build stability	DevOps enables more stable builds.
Bug detection	DevOps enables earlier detection of bugs.
Quality assurance	DevOps enables improved quality assurance.
More	DevOps enables more features to be

features	implemented.
Better testing	DevOps allows for better testing.
Constructive Failure Reviews	DevOps allows for constructive feedback during failure reviews.
Developer well-being	DevOps allows for improved well-being and reduced stress of developers.

Table 4. DevOps Challenges

DevOps Challenge [2, 21, 23, 24, 28, 38, 39, 41]	
Capable members	Getting DevOps capable members into a team is a challenge for us.
Management support	Obtaining management support for DevOps practices is a challenge for us.
Build Process	Increasing complexity of the build process in DevOps is a challenge for us.
Legacy Support	Achieving compatibility between DevOps and legacy systems is a challenge for us.
Pipeline security	Achieving a secure DevOps development process is a challenge for us.
Experienced support	Finding experienced professionals to support DevOps practice is a challenge for us.
Deep seated culture	Changing deep-seated company culture to support DevOps adoption is a challenge for us.
Environment replication	Replicating complex technology environments needed for DevOps is a challenge for us.
DevOps technology	Implementing DevOps technology is a challenge for us.
Tool usage	Using DevOps tools is a challenge for us.
Production access	Access to production systems needed by DevOps is a challenge for us.
Adapting processes	Adapting organisational processes to DevOps is a challenge for us.
Collaboration	Achieving collaboration between Development and Operations is a challenge for us.
Mindsets	Adapting mindsets to achieve successful DevOps is a challenge for us.
Hidden costs	Identifying hidden costs associated with DevOps adoption is a challenge for us.

## 5. Empirical results

At the end of data collection, 82 software development professionals had participated in the study. Two responses were however eliminated due to missing data, yielding 80 useable responses. Demographics of respondents are presented in Table 5. Respondents had spent an average of 8.09 years in the software industry and reported appropriate job roles in areas of software development and IT operations and management.

To address our four research questions we present results of descriptive statistical analysis. Figure 2 depicts the percentages of respondents reporting that they often or always used DevOps practices (RQ1). The

figure also shows the average frequency score for each practice. The results follow a pattern with continuous integration most frequently used, followed by continuous testing. Just over half the respondents (54%) reported frequent use of continuous deployment as a DevOps practice.

Table 5. Sample Profile

Demographics	Category	Frequency	Percentage
Gender	Male	69	86.25
	Female	5	6.25
	Prefer not to identify	6	7.50
Education	High school	17	21.25
	Bachelors	37	46.25
	Postgraduate	21	26.25
	Diploma	3	3.75
	Preferred not to say	2	2.50
Job Role	Developers	58	72.5
	Software Project Managers	5	6.25
	IT Directors / Executives	4	5.00
	Operations	4	5.00
	Analysts	3	3.75
	Consultants	3	3.75
	Testers	2	2.50
	Product Owner	1	1.25

Figure 3 shows the implications of DevOps for software development and deployment culture (RQ2). Over three quarters of respondents agreed that DevOps brought a greater awareness of operations among developers. Over two thirds agreed that DevOps was creating a culture of shared responsibility where developers were taking more responsibility for quality assurance and deployment.

Figure 4 illustrates the benefits of DevOps as experienced by our respondents (RQ3). More frequent builds, earlier detection of bugs and more stable builds are the top three benefits. Most respondents also find improvements to project lead time, easier deployment processes and better feedback.

Figure 5 ranks the challenges of DevOps from highest to lowest (RQ4). Finding capable and experienced DevOps practitioners are the biggest challenges along with changing culture. Implementing DevOps technologies and using automation tools are the least reported challenges.

We confirmed scale reliability and calculated composites scores to represent the extent of DevOps practice ( $\alpha=0.72$ ) and DevOps culture ( $\alpha=0.66$ ) and correlated these with DevOps benefits ( $\alpha=0.83$ ) and challenges ( $\alpha=0.90$ ). DevOps culture was positively

related to benefits ( $r=0.390$ ,  $p<0.01$ ), suggesting that expected benefits are unlikely to materialise unless supported by a DevOps culture. Negative correlations of DevOps challenges with culture ( $r=-0.247$ ,  $p<0.05$ ) and practices ( $r=-0.465$ ,  $p<0.001$ ) suggests that a correctly implemented DevOps culture may result in fewer challenges faced during adoption and usage of DevOps. DevOps practices are also more likely to become routinely used in the absence of challenges. The correlation between the DevOps CI/CD practices and individuals benefits were not all significant. CI was significant for reducing project lead times ( $p<0.05$ ) and reducing stress of developers ( $p<0.05$ ), while more features could be implemented with continuous testing ( $p<0.05$ ). Additional research is needed to better understand how specific DevOps practices translate into benefits. The correlation between DevOps practices and culture was significant ( $r=0.232$ ,  $p<0.05$ ) but not strong. This suggests that there is some unevenness in the degree to which DevOps methods and cultures are being implemented. Organisations might implement certain practices without shifting culture or may be changing culture without succeeding with methods and tooling.

## 6. Discussion

Our research examined DevOps practices and cultures along with perceptions of benefits and challenges experienced in industry. In response to RQ1, results show that continuous integration is the most frequently employed practice with 84% of respondents using CI tooling often or always. Continuous testing was also a common practice with 69% of respondents use automated testing tooling often or always. This suggests a relatively high adoption of build pipelines and CI usage along with the benefits of protection against code decay and regression failures [7, 42]. Continuous deployment was not as frequently practiced within the sample (54%). This suggests that most respondents have yet to experience the higher levels of DevOps maturity [43]. Project managers may also choose not to employ continuous deployment strategies due to risks of faulty builds being deployed to externally exposed environments. Several issues have been raised with the adoption of automated deployments such as a lack of quality and trust within the test suite, deployment being controlled by business decisions and lack of efficient rollback mechanisms [36]. Continuous monitoring usage within the sample was also fairly low. This level of automation is expensive and complex to integrate into production systems [43]. Monitoring can relay uptime, user statistics or testing results in real time, while more complex systems may automatically isolate security breaches and remediate known vulnerabilities [43]. Use of static analysis tooling was relatively low (40%)

within the population, but the practice of continuous testing may be inclusive of static analysis practices.

Our second research question explores how DevOps is influencing software culture. Respondents report that their DevOps culture is largely centered on making developers more aware of operation faults (78%), and having them take greater responsibility for quality assurance (69%) and deployment (68%). A goal of DevOps is to address disconnects between development and operations functions where developers are seen to throw code “over the wall” and defer operational quality and responsibility to operation units [4, 7]. Interestingly, while 60% of respondents agreed that better communication was a part of DevOps culture, less than half the sample agreed that DevOps was supporting a culture of operation representatives attending development meetings. This contrasts with the notion that attendance by both development and operational units in each unit’s meetings is mandatory to improve collaboration between the units [4].

In response to questions on DevOps benefits, 84% agreed that DevOps enables more frequent builds. This finding is consistent with prior work (e.g. [23]). A need to increase the frequency of builds is thus a strong motivation for adopting DevOps. Another strongly reported benefit of DevOps is earlier detection of bugs (80% agree). A strength of a CI pipeline is that development team members are instantaneously notified of build errors. The remaining top five benefits were increased build stability (70%), improved quality assurance (70%) and a reduced project lead time (69%). Reduced lead times are likely due to the usage of deployment pipelines and automation tools [12, 21, 23]. Deploying changes without human intervention is seen as a benefit by nearly two-thirds of respondents (65%). However, continuous deployment was not a frequent practice and therefore not all developers want, or can, implement automated deployment without intervention.

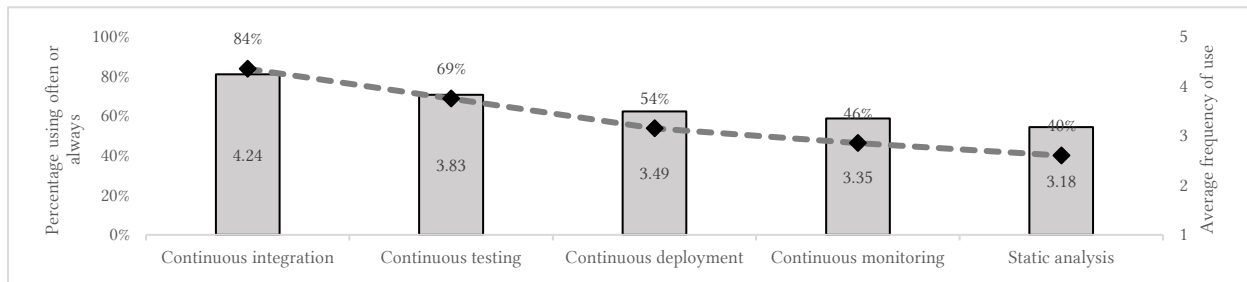


Figure 2. DevOps Practices

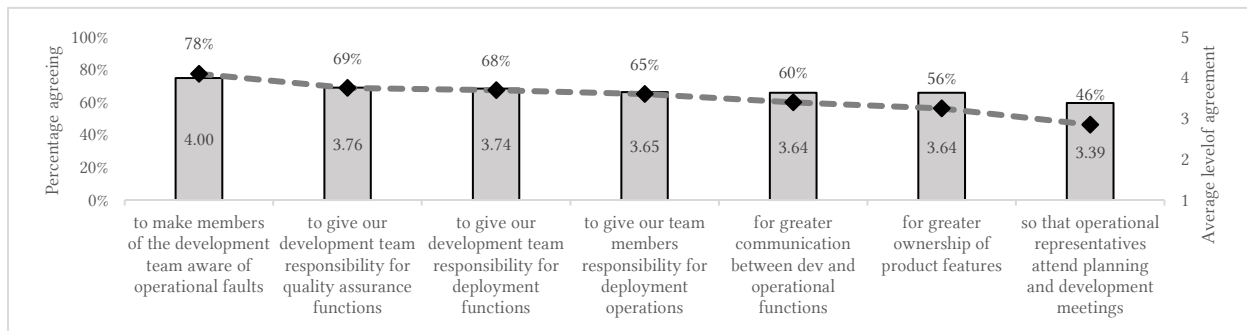


Figure 3. Culture of DevOps

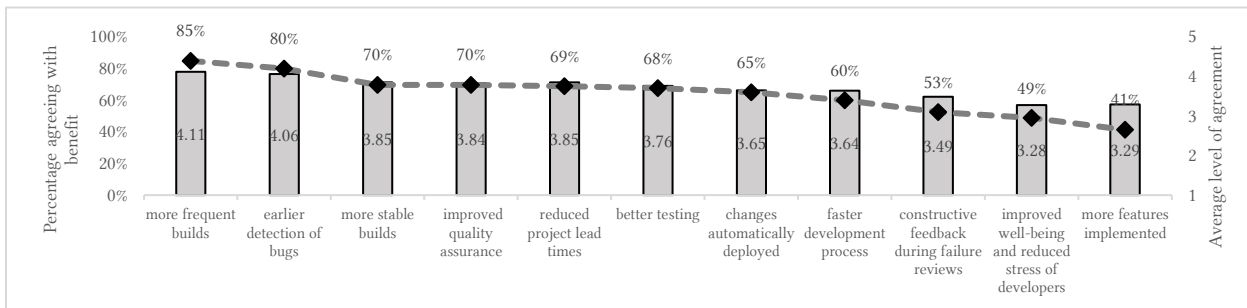


Figure 4. Benefits of DevOps

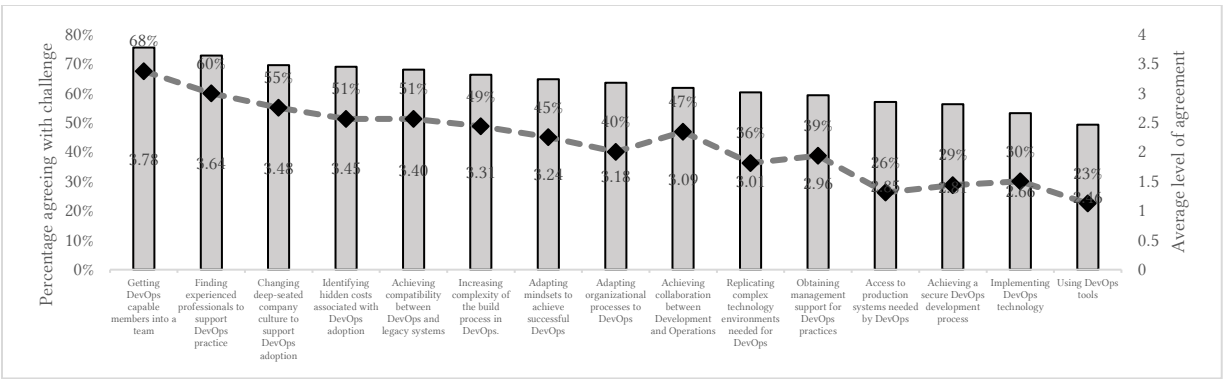


Figure 5. Challenges of DevOps

In line with past work [11], but contrary to others [10], fewer than half our respondents agreed that DevOps reduced stress among developers. Also, only 41% of respondents agreed that DevOps increased their number of implemented features. While DevOps is expected to provide more time for implementing additional features [23], it appears that in our sample more frequent releases are not necessarily about adding more features.

Finally, we focused on DevOps challenges. The top challenges are non-technical and associated with skills and culture. The top two challenges related to a lack of DevOps capable members and difficulty finding experienced DevOps professionals. Lack of skills can result in other software development roles filling DevOps positions [24], which can lead to other challenges. Our sample downplayed technical challenges and generally disagreed that implementing DevOps technologies and tools was a challenge. While it was expected this may be an initial barrier to adoption, many of the tools for a DevOps pipeline are easily accessible as web applications through cloud service providers (e.g. Azure Devops). These tools are generally easy to learn and use, which shifts the DevOps challenge away from tools toward skills and culture. More than half the respondents found culture a challenge for DevOps.

Correlation analysis shows the importance of DevOps culture to realizing benefits and reducing challenges. Results show that automation and CI/CD practices are not by themselves sufficient to achieve DevOps benefits [22]. We thus confirm suggestions made elsewhere that DevOps is not only a technical practice but also a culture [25], and that organisations should not expect the implementation of DevOps continuous practices to automatically translate into benefits without a commensurate culture change [30]. A DevOps culture that removes silos between development and operations is necessary for delivery of DevOps benefits [10, 22].

## 7. Limitations and Conclusion

Our results are limited to analysis of a sample of 80 software developers in the South African context. Sampling has implications for external validity and our results may not generalise beyond our study context. We relied on self-reports and therefore data is subject to response and methods biases. Despite these limitations, our paper has contributed to literature on DevOps experiences, especially in a developing country context. Results show that more frequent builds, earlier detection of bugs and reduced project lead times were the top three benefits, while getting DevOps capable members into a team, finding experienced professionals to support DevOps practice and changing deep-seated company culture to support DevOps were the top three challenges. Continuous integration is the most frequently adopted practice but continuous deployment is not used often. DevOps improves software culture and communication and makes developers more aware of operations. However, ensuring DevOps capable team members and closing skills gaps will be an important priority for industry. Organisations must focus on DevOps culture and not expect CI/CD practices on their own to be sufficient for benefits. Future work may wish to further consider the link between DevOps and outcomes such as stakeholder satisfaction. The costs of implementing DevOps can also be compared against its benefits. Future work might wish to explore CI/CD further and the factors influencing adoption of automated deployment practices. We did not find a majority of respondents agreeing DevOps allowed for more features to be released. Future work should explore the conditions under which this anticipated benefit of DevOps is more likely to be realized. Future work should also explore the impacts of DevOps on IT professional stress, motivation and related job outcomes.



## 8. References

- [1] S.N. Foley, “The Specification and Implementation of “Commercial” Security Requirements including Dynamic Segregation of Duties”. In *4th ACM Conference on Computer and Communications Security*, ACM, 1997, pp. 125-134.
- [2] L. Bass, R. Holz, P. Rimba, A.B. Tran, and L. Zhu, “Securing a deployment pipeline”. In *Third International Workshop on Release Engineering*, IEEE Press, 2015, pp. 4-7.
- [3] A. Hemon, B. Lyonnet, F. Rowe, and B. Fitzgerald, “From Agile to DevOps: Smart Skills and Collaborations”, *Information Systems Frontiers*, <https://doi.org/10.1007/s10796-019-09905-1>. 2019
- [4] J. Humble, and J. Molesky, “Why Enterprises Must Adopt DevOps to Enable Continuous Delivery”, *Cutter IT Journal*, 24(8), 2011, pp. 6-12.
- [5] A. Wiedemann, N. Forsgren, M. Wiese, H. Gewalt, and H. Krömer, “Research for Practice: the DevOps Phenomenon”, *Communications of the ACM*, 62(8), 2019, pp. 44-49.
- [6] L. Leite, C. Rocha, F. Kon, D. Milojicic, and P. Meirelles, “A Survey of DevOps Concepts and Challenges”, *ACM Computing Surveys (CSUR)*, 52(6), 2019, pp. 1-35.
- [7] B. Fitzgerald, and K.J. Stol, “Continuous Software Engineering: A Roadmap and Agenda”, *Journal of Systems and Software*, 123, 2017, pp. 176-189
- [8] R. Jabbari, N. bin Ali, K. Petersen, and B. Tanveer, “What is DevOps? A Systematic Mapping Study on Definitions and Practices”. In *Scientific Workshop Proceedings of XP2016*, 2016, pp. 1-11.
- [9] M. Walls, *Building a DevOps Culture*, O'Reilly Media, 2013.
- [10] M. Senapathi, J. Buchan, and H. Osman, “DevOps Capabilities, Practices, and Challenges: Insights from a Case Study”. In *22nd International Conference on Evaluation and Assessment in Software Engineering*, 2018, pp. 57-67.
- [11] J. Shropshire, P. Menard, and B. Sweeney, “Uncertainty, Personality, and Attitudes toward DevOps”. In *Twenty-third Americas Conference on Information Systems*, Boston, 2017.
- [12] A. Wiedemann, “A New Form of Collaboration in IT Teams-Exploring the DevOps Phenomenon”. In *PACIS*, 2017, paper 82.
- [13] L.E. Lwakatare, T. Karvonen, T. Sauvola, P. Kuvaja, H.H. Olsson, J. Bosch, and M. Oivo, “Towards DevOps in the Embedded Systems Domain: Why is it so Hard?”. In *49th Hawaii International Conference on System Sciences*, IEEE, 2016, pp. 5437-5446.
- [14] A. Wiedemann, and M. Wiese, “Are You Ready for DevOps? Required Skill Set for DevOps Teams”. In *Twenty-Sixth European Conference on Information Systems*, 2018, Paper 123.
- [15] M. Rajkumar, A.K. Pole, V.S. Adige, and P. Mahanta, “DevOps Culture and its Impact on Cloud Delivery and Software Development”. In *International Conference on Advances in Computing, Communication, & Automation*, IEEE, 2016, pp. 1-6.
- [16] B. Violino, “Real-world Devops Failures--and How to Avoid Them”, InfoWorld.com, 2016, available online: <https://www.infoworld.com/article/3087447/real-world-devops-failures-and-how-to-avoid-them.html>
- [17] Puppet.com “2017 State of DevOps Report”, 2017, available online: <https://puppet.com/resources/whitepaper/2017-state-of-devops-report>
- [18] L.E. Lwakatare, “DevOps Adoption and Implementation in Software Development Practice: Concept Practices Benefits and Challenges”, Ph.D. dissertation, University of Oulu, 2017.
- [19] Gartner. “Gartner Forecasts IT Spending in South Africa”, 2019, available online: <https://www.gartner.com/en/newsroom/press-releases/2019-07-24-gartner-forecasts-it-spending-in-south-africa-will-gr>
- [20] A. Schofield, and B. Dwolatzky, “2019 JCSE-IITPSA ICT Skills Survey”, 2019,. Available online: <https://www.iitpsa.org.za/jcse-iitpsa-skills-survey/>
- [21] V. Gupta, P.K. Kapur, and D. Kumar, “Modeling and Measuring Attributes Influencing DevOps Implementation in an Enterprise Using Structural Equation Modeling”, *Information and Software Technology*, 92, 2017, pp. 75-91.
- [22] W.P. Luz, G. Pinto, and R. Bonifácio, “Building a Collaborative Culture: a Grounded Theory of Well Succeeded DevOps Adoption in Practice”. In *12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 2018, pp. 1-10.

- [23] L. Riungu-Kalliosaari, S. Mäkinen, L.E. Lwakatare, J. Tiihonen, and T. Männistö, “DevOps Adoption Benefits and Challenges in Practice: A Case Study”, In *International Conference on Product-Focused Software Process Improvement*, Springer, 2016, pp. 590-597.
- [24] A.S. Amaradri, and S.B. Nutalapati, “*Continuous Integration, Deployment and Testing in DevOps Environment*”, Thesis, Blekinge Institute of Technology, Sweden, 2016.
- [25] S. Jones, J. Noppen, and F. Lettice, “Management Challenges for DevOps Adoption within UK SMEs”. In *2nd International Workshop on Quality-Aware DevOps*, ACM, 2016, pp. 7-11.
- [26] M.B. Kamuto, and J.J. Langerman, “Factors Inhibiting the Adoption of DevOps in Large Organisations: South African Context”, In *2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology*, IEEE, 2017, pp. 48-51.
- [27] F.M.A. Erich, C. Amrit, and M. Daneva, “A Qualitative Study of DevOps Usage in Practice”, *Journal of Software: Evolution and Process*, 29(6), 2017, e1885.
- [28] J. Hamunen, “Challenges in Adopting a DevOps Approach to Software Development and Operations”, Dissertation, Aalto University, Finland, 2016.
- [29] A.A. Khan, and M. Shameem, “Multicriteria Decision-Making Taxonomy for DevOps Challenging Factors Using Analytical Hierarchy Process”, *Journal of Software: Evolution and Process*, 2020, e2263.
- [30] K. Kuusinen, V. Balakumar, S.C. Jepsen, S.H. Larsen, T.A. Lemqvist, A. Muric, ... and O. Vestergaard, O. “A Large Agile Organization on Its Journey towards DevOps”. In *44th Euromicro Conference on Software Engineering and Advanced Applications*, IEEE, 2018, pp. 60-63.
- [31] A.M. Wiedemann, and T. Schulz “Key Capabilities of DevOps Teams and their Influence on Software Process Innovation: a Resource-Based View”. In *Twenty-third Americas Conference on Information Systems*, Boston, 2017.
- [32] A. Wiedemann, M. Wiesche, and H. Kremer, “Integrating Development and Operations in Cross-Functional Teams-Toward a DevOps Competency Model”. In *Computers and People Research Conference*, 2019, pp. 14-19.
- [33] M. Hilton, N. Nelson, T. Tunnell, D. Marinov, and D. Dig, “Trade-offs in Continuous Integration: Assurance, Security, and Flexibility”. In *11th Joint Meeting on Foundations of Software Engineering*, ACM, 2017, pp. 197-207.
- [34] N. Nagappan, and T. Ball, “Static Analysis Tools as Early Indicators of Pre-Release Defect Density”. In *27th International Conference on Software Engineering*, ACM, 2005, pp. 580-586.
- [35] E. Laukkanen, J. Itkonen, and C. Lassenius, “Problems, Causes and Solutions when Adopting Continuous Delivery—A Systematic Literature Review”, *Information and Software Technology*, 82, 2017, 55-79.
- [36] M. Shahin, M.A. Babar, M. Zahedi, and L. Zhu, “Beyond Continuous Delivery: an Empirical Investigation of Continuous Deployment Challenges”. In *11th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, IEEE, 2017, pp. 111-120.
- [37] M. Sánchez-Gordón, and R. Colomo-Palacios, “Characterizing DevOps Culture: a Systematic Literature Review”. In *International Conference on Software Process Improvement and Capability Determination*. Springer, 2018, pp. 3-15.
- [38] M.A. McCarthy, L.M. Herger, S.M. Khan, and B.M. Belgodere, “Composable DevOps: Automated Ontology Based DevOps Maturity Analysis”. In *IEEE International Conference on Services Computing*, IEEE, 2015, pp. 600-607.
- [39] Y. Liu, and Y. Zhou, “*The Challenges and Mitigation Strategies of Using DevOps during Software Development*”, Dissertation, Blekinge Institute of Technology, Sweden, 2017.
- [40] J. Roche, “Adopting DevOps Practices in Quality Assurance”, *Communications of the ACM*, 56(11), 2013, pp. 38-43.
- [41] B.S. Farroha, and D.L. Farroha, “A Framework for Managing Mission Needs, Compliance, and Trust in the DevOps Environment”. In *Military Communications Conference*, IEEE, 2014, pp. 288-293.
- [42] D. Saff, and M.D. Ernst, “An Experimental Evaluation of Continuous Testing During Development”, *ACM SIGSOFT Software Engineering Notes*, 29(4), 2004, pp. 76-85.
- [43] Puppet.com “*2018 State of DevOps Report*”, 2018, Available online: <https://puppet.com/resources/whitepaper/state-of-devops-report>