Testing in a DevOps Era: Perceptions of Testers in Norwegian Organisations

Daniela Soares Cruzes[1], Kristin Melsnes[2], Sabrina Marczak[3]

[1] SINTEF - Norway
[2] Bouvet – Norway
[3] PUCRS - Brazil
danielac@sintef.no, Kristin.melsnes@bouvet.no, sabrina.marczak@pucrs.br

Abstract. To better understand the challenges encountered by testers in DevOps development, we have performed an empirical investigation of what are the trends and challenges for the testers in the DevOps environment. We have discussed the quality assurance in the difference focus areas of DevOps: Social Aspects, automation, leanness, sharing, measurement. The results were then themed in five different topics of concern to testers: collaboration, roles and responsibilities, types of tests, automation and monitoring and infrastructure. In Testing, there has been a change on the roles and responsibilities of testers, where there is much more focus on the responsibilities for testing across the teams, instead of a sole responsibility of the tester. Testers are also forced to collaborate more with other stakeholders as operations and business. Testing is brought to another level of automation in DevOps but there is still need for manual tests, that have to be much more risk-based than before. And finally, testing transparency is a must in this process and should involve not only development team but also operations and customers. This paper contributes to the body of knowledge on what are the areas we need to focus for improvement in testing for the DevOps environment. This paper also contributes to practitioners to improve their testing focusing on specific areas that needs attention.

Keywords: DevOps, Testing, Empirical.

1 Introduction

DevOps is a term that has been increasing in popularity since 2009. DevOps emerged from continuous software delivery, which captures market opportunities and reduces feedback time. DevOps is a movement stemming from practice seeking to address certain limitations of agile methodology in terms of operational and business readiness. More specifically, this approach aims towards a shorter time to market [6]. Debois [10] opined that the term DevOps is just a stub for more global company collaboration, which, he explains, works as follows: Once priorities have been determined and work can begin, developers pair together with operations people to get the job done. This pairing allows for better knowledge diffusion across the two traditionally separate teams. Issues such as stability, monitoring and backup can be addressed immediately, rather than being afterthoughts, and operations staff obtain a better understanding of how the application works before it is deployed to production. In addition, feedback is available to all staff: those in operations learn which issues they might expect in production, while developers learn about the production environment. Gottesheim [16] affirms that DevOps aligns business requirements with IT performance, with the goal of adopting practices that allow a fast flow of changes to a production environment, while maintaining a high level of stability, reliability and performance in these systems.

While a central goal of DevOps is to achieve an improved deployment frequency, it is crucial to consider how QA plays a role in helping or hindering that ability. An important driver for DevOps is the need for businesses to quickly enact changes to the market. At the same time, quality must remain high to, for example: i) attract new and retain existing customers and users and offer professional systems; ii) avoid downtime; and iii) provide user-friendly solutions and low support. In this context, software testing is not only 'essential', but is 'critical' for the software application or product survival.

However, throughout the history of software development, testing techniques have struggled to keep up with the ever-changing trends in software development paradigms, and little data exists on software professionals' software testing and quality assurance (QA) practices [18]. In an agile environment, continuous testing is mostly viewed as focusing on testing as early as possible, whereas in DevOps it is seen as a continuous activity, which is also performed in production. Whether developing, testing, or deploying software, DevOps seeks the most efficient mode of delivering an application continuously through the pipeline. This leads to a requirement for significantly more collaboration and effective cross-functional teams, where everyone is responsible for quality. Furthermore, a team works towards less documentation and increased transparency in testing activities.

Angara et al. [1] performed a systematic review of 29 articles concerning factors driving testing in the DevOps setting, to understand the different motivational factors and attempts to identify key technical, cultural and managerial factors behind testing in the DevOps setting. The authors observed that testing in DevOps is closely associated with the automation of test cases. Factors such as agility, scale, metric driven processes, and reduction of complexity and costs appeared in more than 16 review articles. DevOps demands alternative metrics, to achieve better collaboration and communication between various stakeholders of the system. Angara et al. [1] concluded that DevOps testing has not been systematically studied in academic scientific literature. Angara et al. [1] did not find any real-time case studies in the context of DevOps testing frameworks in academic journals. The traditional isolated QA/testing skills may be challenged and play a limited role in DevOps. A QA/tester has to scale beyond regular testing functions, and aid in development and operations teams to satisfy the philosophical objectives of DevOps. DevOps testing requires the design of alternative metrics/measures, which elevate the culture of an organisation, increase collaboration and create a proper benchmarking base.

In this study, we expand on previous work by focusing on QA/testing in DevOps development. We aim to obtain a further understanding of what can be done to improve teamwork between developers and testers in a DevOps testing context. We interviewed testers, with a focus on the following research questions:

- What is new for testing in DevOps?
- Which approaches for testing are utilised in DevOps?
- What are the challenges for testing in DevOps?

The remainder of this paper is organised as follows. Sections 2, 3 and 4 recapitulate the DevOps principles. Section 5 describes the research methodology followed in this research, and Section 6 presents the results. In Section 7, we discuss the results in light of previous literature, as well as the implications for research and practices concerning DevOps testing. Finally, Section 8 concludes this paper.

2 DevOps Principles

In 2016, França et al. [13] described a set of principles, skills and tools to support the practices characterising DevOps based on the DevOps literature. **Fig. 1** illustrates the principles associated with DevOps, grouped into six categories, which are briefly explained as follows [13]:

- Automation: claimed to be one of the core principles of DevOps, owing the benefits it could promote. It is considered that manual and repetitive tasks can be automated to reduce unnecessary effort and improve software delivery. Hence, automation would improve not only the delivery speed, but also the infrastructure consistency, productivity of teams and repeatability of tasks.
- Quality Assurance: to assure the quality of both development and operations processes as products. This principle supports the implementation of DevOps practices, as it links different stakeholders (development, operations, support and customers) to perform activities in an efficient and reliable manner, as well as to achieve a product and services satisfying established quality standards.
- Leanness: some DevOps practices are based on lean thinking principles [26]. DevOps requires lean processes, as it intends to ensure a continuous flow to develop and deliver software regularly through small and incremental changes, thus fostering constant and fast feedback between the development, testing and operations teams, as well as with customers.
- Sharing: information and knowledge are disseminated among individuals to promote the exchange of personal learning and project information. In this sense, individuals should spread relevant information, for instance regarding how to implement and perform practices recommended in the context of DevOps.
- Measurement: an important principle often instantiated by collecting efficient metrics to support decision-making in the software development and operations lifecycle.
- Social Aspects: despite all technical principles, many DevOps characteristics are associated with social aspects among the software development and operations teams, which is one of the reasons that DevOps is often seen as a cultural shift.

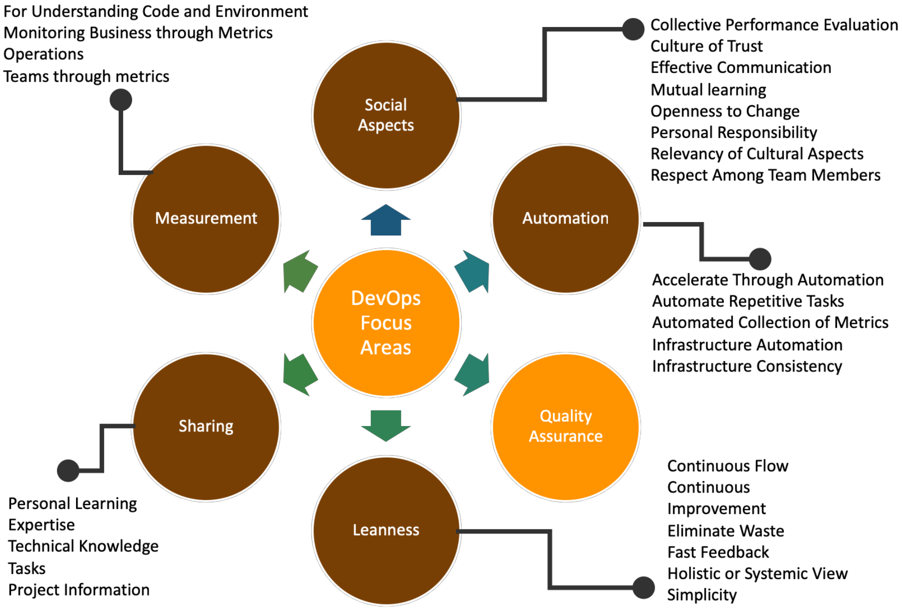

Fig. 1. DevOps principles by França, Junior and Travassos [13]

3 Research Methodology

In the study presented here, we aim to investigate the testing process for DevOps projects. To answer our research questions, we performed various data collection methods. Initially, as there were no scientific papers to be found, we focused on understanding the insights we could glean from various blogs and posts online concerning DevOps Testing. The characteristics of these source are that information is scattered, with each blog focusing on different aspects of DevOps principles, and most focusing only on automation aspects, while others focus on culture or the shift-left needs. These blogs were used to create a interview guide for discussion with testers.

The context of this study concerns a consultancy firm providing services in information technology, digital communication and enterprise management to different private and public organisations in Norway. Bouvet have approximately 1300 employees at 14 offices in Norway and Sweden. Furthermore, Bouvet is committed to maintaining long-term client relationships, and is a strategic partner for a number of enterprises through innovation, development and implementation of solutions of critical significance for society.

Bouvet’s approach is adapted to the customer, culture, directions and contract. The testers are all experienced testers and are allocated to different projects each year. The testers are allocated to different companies, and therefore they express experiences of different environments. There are 15 testers. For this study, there were multiple discussions with the test leader. In addition, the first author conducted one-on-one interviews with 11 of the testers (See Table 2 for the interview guide). Each interview lasted for approximately one hour. As preparation for the interview, the testers were asked to provide a PowerPoint presentation with four slides, in which they would begin to answer the following four questions:

1) How do you perform agile testing today?
2) What works well?
3) Which challenges do you foresee during the next year in your role in DevOps teams?

After the semi-structured interviews (Table 2), we ran a full day meeting with all testers. The meeting consisted of each tester presenting on different aspects of testing for DevOps, and the first author of this paper presented the results from the interviews in the form of a validation of the analysis of the results from the interviews. Suggestions and comments were then gathered, and incorporated into the presentation of the results to a larger audience at a testers' conference in Oslo for 250 testers, and in DevOpsDays Oslo, where the feedback was also very positive (the audience consisted mainly of developers). Three other presentations of the results were delivered to different companies in Norway, where the results were again positively accepted.

Table 1. Data collection

	Data Collection
Focused interview	Test Leader/ 11 Testers
Blogs, magazines and online reports	✓
Meetings with testers	✓
Interviews	✓
Plenary presentations to testers and developers	✓

Table 2. Interview guide

INTERVIEW GUIDE

1. Would you say that you are performing agile testing today? Can you describe what you do today?
2. What is DevOps Testing for you?
3. What is the difference from agile testing in terms of processes and approaches? How do the different types of tests change, such as unit testing and functional testing?
4. What is different from agile testing in terms of collaboration and communication with ops and developers? Can you describe how communication between dev and ops teams usually occurs within your projects?
5. What kind of information do you usually receive/send to/from the other teams? Could you provide some examples? How often does this communication occur? How long does it take? How do you think this will be different with DevOps testing?
6. Are there meetings in which all teams are invited to participate (product release, etc.)? How do you think these will change with DevOps?
7. In terms of automation, do you think things will change much from what you do today?
8. Do you perform any types of monitoring activities today that help with testing? Do you foresee any possible monitoring activities that will be introduced by DevOps testing?
9. What is the "infrastructure as code" for testing? How do you see this topic?
10. Do you foresee any changes in your role/competence or responsibilities in this new scenario of performing DevOps testing?Are there barriers to DevOps testing, such as individual (skills and competences), team (work in teams), project characteristics (small, big, public, private or fixed contract) or organisational (impossible to change structures) barriers?
11. What are the major enablers in achieving DevOps testing?

4 Results

Table 3 presents a summary of the benefits that testers mentioned by improving testing practices in the DevOps environment. The table also presents the strategies mentioned by testers to improve testing in different areas. In this section, we also summarise the results, based on an analysis of the interviews, and other insights collected concerning the different observations of the testers' discussions on the topic with pratictioners in Norway.

4.1 Collaboration

In agile projects, testers and developers have already been brought much closer together: testers have left 'tester teams' to become part of the 'development team', where testers are allocated to each development team in an agile team. There is already a positive culture of open and effective communication. In DevOps, the testers' perspective is that they are 'forced' to also become closer to operations and business teams. The perception of testers is although that there is more pressure to be closer to business than operations.

To be able to achieve continuous deployment, testers have to be much more closely aligned to what the business is expecting of the deliverables. In some teams, customers have their own representatives in the team. However, it is challenging to gain an understanding from both developers and customers to participate in and execute early testing at short notice. This demands a high level of flexibility, understanding and attendance from the testers' perspective. There is also a requirement to build trust and transparency in the testing process for customers, such as in collaborating with users on test automation. One challenge mentioned regarding the relationship with customers is that they are not always prepared for a tighter communication stream with development, and therefore collaboration is hindered.

In projects worked on by the testers, testers still find that communications between testers and operations (OPS) are ad-hoc. Specifically, when there is a separate operation department, collaboration is even harder. From the perspective of the testers, one of the main reasons for challenge is that the OPS often also has other reward systems and incentives, and is therefore opposed to DevOps goals.

4.2 Roles and Responsibilities

The testers mentioned that there is a requirement for teams to rethink their roles and responsibilities towards QA activities. The focus is more on competences than on roles, but one tester mentioned: '*There is still a need to find a leader/facilitator/advisor for testing, because the whole team approach needs to be better understood'*. The testers mentioned that the following competences are increasingly requested of testers:

- More technical understanding (code; architecture; coding; and infrastructure such as virtualisation and operations);
- More knowledge on test automation;
- Different testing techniques, such as security testing;

Table 3. Benefits of improvements for DevOps and strategies

	Benefits of improvements for DevOps	***Strategies used by teams***
Collaboration	More collaboration with operations: • Helps on obtaining more knowledge about the system and how it should work. • Provides more complete information to the test. • Helps testers to find operating errors earlier. • Facilitates the handover process. • OPS traditionally have more experience with monitoring, logging of activities; testers can learn from them. • Better test coverage by everyone thinking about quality. • Closer proximity to the production environment.	• Collaborate with users on writing test acceptance criteria on user stories. • Add testers to review user stories as part of the definition of "prepared". • Give access to automated test results to customers. • Explain the team's goals on quality to all stakeholders. • Increase testing process transparency to al stakeholders. • Tighter the communication with customers regarding testing.
Roles and Responsibilities	Testers mentioned that the benefits of having developers increasingly involved with different testing activities include: • By involving developers in the specification and test process earlier, testers gain greater knowledge of the business and solution, consequently preventing errors by building better solutions. • Bugfix is the most 'boring' task for developers. By contributing to finding deviations early, developers will spend less time on error correction. • Creating a culture where 'finding mistakes is positive'. Avoid tab quotes and blame games. Celebrating when we finding errors before the system is in production. .	• Pair testers with developers and operations for testing activities. • Job rotation: testers as developers and developers as testers. • Post-mortem analysis of bugs in production. • Learning from bugs through statistics. • Celebrating bugs. • Bug crush day: having a day when everyone in the team is only working on closing bugs. • Encourage team members on learning of testing competences ('learning the testing language').
Types of Testes	• The process becomes less test-case oriented more risk-based. • Testing becomes more about minimising the risk of critical errors in production, and continuously assessing the impacts on costs related to testing Automation of tasks to prevent human error becomes increasingly essential to the process.	• Risk-based acceptance and regression tests (scenario-based and customer-based). • Take risks and evaluate the impact of changes in real time (e.g., A/B testing). • Experimentation of features. • Feature toggle. • Testing in production.

	Benefits of improvements for DevOps	*Strategies used by teams*
Automation	Manual test will basically complement automated tests: • Dynamic, more unplanned and unpredictable. • Focus on tests that requires Knowledge, creativity and special situations that are expensive or not practical to automate, user experiences and shopping patterns are still difficult to automate.	• Proper Exploratory tests.
Monitoring	Considerable improvement is required in this area. However, there is not much experience reported here.	• Bug monitoring. • Green-light test cases. • Measurements of usage of features.

- More business and strategic thinking:
  - Risk-based testing;
  - Manual tests that focus on the customer's quality goals;
- Playing the role of advisor/leader/facilitator for testing activities:
  - Following up that testing is performed continuously and has good coverage.
- Learning techniques for testing in production.

The testers also recognise that some 'traditional' behaviours should be discarded, such as '*having a black box testing only mind set*'. Furthermore, some old skills need to be reinvented, such as adapting lean practices of demand management, governance, standardisation and optimisation techniques; and delivering pre-packaged test suites, estimation models and risk-based testing with the aim of decreasing development before deployment time.

Concerning the question of whether there is still a need for someone in the team to be named the tester, the testers opined: '*Testers' skills will still be important: we have our own ability to dig out errors, and we have a great deal of knowledge about techniques that can be used to bring them about*'. Another tester mentioned: '*We must build trust and get the team to trust our knowledge*'.

There is also a need for the team to have an end-to-end responsibility for the product. This means that the entire team is involved in the testing/QA of the delivered content. Testers mentioned that *'developers are generally not used to testing functionality/user experience: they basically do not want additional tasks'*.

The testers also see the need to develop the testing skills of developers, so that there is less of a bottleneck on testing activities in the continuous flow of delivered features. Testers mentioned that they expect that developers will *'perform some functional tests, include tests in the definition of complete and in code reviews, and perform their part of testing'*.

4.3 Types of Tests

In DevOps, the focus shifts from the phases and quadrants of testing [8] to an approach that is focused on filtering bugs. Testing becomes is so integrated into the overall development process that it can be difficult to determine who, where and when testing is performed. The main goal is to prevent and eliminate coding errors before the application has reached production. This also calls for more risk-based testing [22,12,11] processes, and less test-case oriented testing. The testing becomes more focused on minimising the risk of critical errors in production, and continuously assessing the impacts on costs related to testing. The automation of tasks to prevent human error becomes increasingly essential in this process. One tester mentioned: *'We need a thought-through streamline test process, and the teams need to have a very clear and defined test process with the steps that each component has to go through before it is released into production'*. Another tester mentioned: *'It's about continuously getting feedback right from the start. And applying the concept of Fail fast, fail often and getting frequent feedback on whether things work or not'*.

The types testing documentation change as well, there is less focus on test case documentation and more on streamlining test specifications. Updating of test scenarios each time something changes is avoided to save time and effort, and there is increased use of exploratory testing and charters showing the areas of testing and types of tests that are performed for each part of the system. One tester mentioned: *'The focus on documentation was previously to build the trust that we have tested what we should have. Now we don't have time for that, and we need to build trust in other ways'*. One approach to this is to have more transparency in the testing process for all stakeholders. One tester mentioned: *'When the team have closer contact with each other documentation lose importance'*.

On the non-functional aspects of the system, security has been more emphasised, and is increasingly demanding as a type of test to be performed and learned. The testers mentioned that this is one of the major obstacles for them to implement DevOps testing. The testers noted that other non-functional requirements are also important, such as scalability, performance and user experience, and that developers are usually mostly focused on testing whether the functionality works.

4.4 Automation

The testers in this study emphasised the importance of understanding that continuous testing is not the same as test automation. Code quality, pipeline automation, application quality and customer experience are all steps towards achieving a higher quality product through continuous delivery. The testers in the interviews emphasised the need for manual testing in most projects. One main message that the testers would like to get across is *'do not ignore the power of manual tests'*.

For the testers, manual testing has its own advantages and disadvantages. In manual testing, the tester can unveil many problems, from user experience issues to critical issues, but it can take a long time to perform and require all of the tester's attention. In manual testing, the testers becomes familiar with the tested application and aware of all its pitfalls. Therefore, by performing manual testing there are more chances to easily explore the system and know exactly where to look for hidden bugs.

One tester mentioned: *'In automation, we robotise the testing. This is very effective for (boring) routine repetitive tasks, interface simulations, combinations of data, loads and so on. However, this are somehow static. Such tests perform again and again what you asked them to do a week ago'*. One tester mentioned that *'teams sometimes forget that automated tests are also code: they require maintenance, they can contain errors and they become outdated'*.

4.5 Monitoring and Infrastructure

Consistent with the observation that testers are still not collaborating as much with operations, testers in this aspect, therefore the testers had less experience to discuss concerning the principles of operations. The main conclusion from this research in this topic is that tools, processes and monitoring data are not yet explored in development. Testers were not able to have an opinion on how monitoring could affect testing.

5 Discussion

5.1 What are the Main Challenges on Testing in DevOps?

DevOps calls for more efficient testing to satisfy customers and end-users requirements. The challenges mentioned most often were:

- Full automation is a danger to the product;
- A whole-team approach to quality is hard to achieve;
- There is still a requirement for increased maturity of the whole development chain for taking full advantage of DevOps.

DevOps tools can significantly automate and streamline development, but they must be integrated into a complete solution that simplifies the entire process. The right test environment management and test automation tools allow teams to increase their test speed and coverage. However, these tools also still require further maturation. As noted previously, manual testing remains a very important activity for QA for many different reasons. One tester mentioned: *'Automated functional testing at all levels works against valuable input to both the team and the customer. It is a challenge to move the team away from the business knowledge and application knowledge they gain when they participate in manual functional testing. I believe that the understanding of the application and customer business is a big part of the high quality in deliveries we have achieved in our team'*.

As previously mentioned in this paper, it is already a challenge in agile to integrate everyone into the testing process. In DevOps, this is even more pronounced, as it requires even more collaboration and alignment. The cooperation with operations is even more challenging, because traditionally the IT infrastructure personnel are distanced from the reality of the development process. From the perspective of testers, developers are still yet to change their mindset towards testing, to think holistically about the implementation of the features where the definition of 'done' includes the whole quality chain that will ensure the feature is ready for production. Testers also mentioned that it has been challenging to gain an understanding from both developers and customers that there is a need for their participation of the tests as early as possible and many times at a short notice.

The motivation of developers towards testing activities represents an additional challenge. The perception of testers is that developers are not very interested in testing, and that testing is considered to be boring. In Norway, there is also a trend that testers do not have development skills, and most testers are used to doing everything manually. Therefore, testers are not focused on test automation. Many teams are working on the process of improving collaboration between testers and developers to create automated tests.

DevOps requires considerable trust and maturity from the all stakeholders (customers, suppliers etc.), and the testers pointed out the need to understand that this will not always be a fit. The companies/teams that wishes to implement DevOps will have to adapt the approach to their contexts and understand which practices can be adopted in their contexts.

5.2 Bridging Practice and the State-of-the-Art in Testing in DevOps

Table 4 lists the implications of the practice of and research on each of the principles shown in Fig. 1.

Table 4. Implications for research

DevOps Principle	*Main Conclusions*	*Implications for Research*
Social Aspects	The whole-team approach to QA is essential.	Some studies indicate that communication processes tend to have the capacity to mediate interactions and misalignment between developers and testers [2, 24, 4, 27]. More research on collaboration and communication must be conducted to help organisations understand how to improve the whole-team approach and interactions between testers and developers, operations, business and other stakeholders.
Automation	Automation is a must, but manual testing still cannot be ignored.	Automation is a hot topic in academia, and there is a vast amount of research on this topic. The challenges and approaches are covered well, but there remain many challenges to be covered [23, 15, 14, 21].
Leanness	The process of QA still requires maturation, and all principles of DevOps should be addressed to get QA working well. There is a need for a more risk-based testing processes and less test-case oriented testing.	More empirical work must to be conducted to understand which approaches work where. We also need to evolve risk-based testing, and achieve wider-spread use in industry [22,12,11]. We need a better understanding of how to achieve a better return of investment of DevOps practices and principles depending on the context. In addition, we need frameworks and theories that support the adoption of DevOps by other parts of organisations, such as higher-level managers and sellers or products/projects.

DevOps Principle	*Main Conclusions*	*Implications for Research*
Sharing	Working as a cross-functional team is a must. Roles and responsibilities in QA are changing, and competences must also change.	We have not yet found research papers focusing on understanding how to get teams to work efficiently as cross-functional teams and using roles, responsibilities, competences in the best manner for cross-functional teams in DevOps QA.
Measurement	There are blogs on test environment management and performance testing as a code. However, not much material is available on measurements that help teams improve their work in testing.	There are discussions in academia about using artificial intelligence to improve the testing of products, but companies are not yet adopting these approaches [19, 5]. In addition, there is the concept of chaos engineering, which can be better explored in DevOps environments [3].

5.3 Limitations

The research reported here has limitations. In terms of reliability of the study, i.e., how dependent are the data and analysis on the researchers involved [25]. First, our study was limited to members of one company. However, the participants are allocated to different projects in different organisations with different characteristics, helping to overcome certain issues concerning the generalisability of our results. There is also the limitation that this study describes the perspective of testers regarding the studied phenomena. To mitigate this limitation, we have performed presentations to developers, and validated the results in that context. We have also used blogs focusing on different roles in different organisations in the results. Finally, the observations and findings were verified with other companies' representatives, to avoid false interpretations and inconsistencies.

Another limitation of our study is that we relied mainly on interviews and observations to derive our results. To mitigate this threat, we discussed and improved the interview protocol iteratively before data collection, besides we have used blogs and academic literature to drive the construction of the interview guide. Concerning the analysis, the researchers coded all interviews individually, and in a second step all authors discussed all the results including the interviewees.

6 Conclusions

In this paper, we have presented the results from a study on testing/QA activities in DevOps. We have also identified challenges, approaches and opportunities in different areas such collaboration, roles and responsibilities, types of tests, automation and monitoring and infrastructure. We could observe that to trigger any improvement in the state of this practice there will be necessary serious efforts in predicting trends, learning stakeholder mindsets, and pinpointing software-testing problem areas. This paper represents a starting point for some of these issues, which require attention from both practitioners and academics as described in the discussion.

As future work we plan to further investigate the trends to DevOps quality assurance and investigate possible new techniques that can be spready adopted by software teams in DevOps development projects.

Acknowledgments

This work was supported by the EMERGE project, funded by the Research Council of Norway under the grant 231679.

References

1. Angara, J.; Prasad, S.; Sridevi, G.. The Factors Driving Testing in DevOps Setting- A Systematic Literature Survey. IJST, Jan. 2017. ISSN 0974 -5645.
2. Barney, S., Mohankumar, V. , Chatzipetrou, P., Aurum, A., Wohlin C. and Angelis, L. "Software quality across borders: Three case studies on company internal alignment" *Inf. Softw. Technol.* 56, Jan. 2014, 20-38.
3. Basiri, A., Behnam, N., Rooij, R., Hochstein, L., Kosewski, L., Reynolds, J. and Rosenthal, C.: Chaos Engineering. IEEE Software 33(3): 35-41 (2016)
4. Bjarnason, E., Runeson, P., Borg, M. , Unterkalmsteiner, M., Engström, E., Regnell, B., Sabaliauskaite, G., Loconsole, A. , Gorschek, T. and Feldt, R.: "Challenges and practices in aligning requirements with verification and validation: a case study of six companies." Empirical Software Engineering 19(6): 1809-1855 (2014)
5. Briand, L.C.,Novel Applications of Machine Learning in Software Testing. QSIC 2008: 3-10.
6. Claps G.G., Svensson R.B, Aurum A. On the journey to continuous deployment: Technical and social challenges along the way. IST. 2015; 57:21–31.
7. Collins, E., Macedo, G., Maia, N., and Dias-Neto, A., "An Industrial Experience on the Application of Distributed Testing in an Agile Software Development Environment," IEEE Seventh International Conf. on Global Software Engineering (ICGSE), 2012, pp. 190-194.
8. Crispin L. and Gregory, J, "Agile Testing: A Practical Guide for Testers and Agile Teams", Addison-Wesley Professional, 2009.
9. Cruzes,, D.S., Moe, N.B., Dybå, T.: Communication between Developers and Testers in Distributed Continuous Agile Testing. ICGSE 2016: 59-68
10. Debois, P., "Devops: A software revolution in the making," Cutter IT Journal, vol. 24, no. 8, pp. 3–5, Aug 2011.
11. Felderer, M., Grossmann, J., Schieferdecker, I.: Recent Results on Classifying Risk-Based Testing Approaches. CoRR abs/1801.06812 (2018).
12. Foidl, H. and Felderer, M.: Integrating software quality models into risk-based testing. Software Quality Journal 26(2): 809-847 (2018).
13. França, B. B. N., Jeronimo Jr., H., Travassos, G.H.: Characterizing DevOps by Hearing Multiple Voices. SBES 2016: 53-62.
14. Garousi, V., Mäntylä, M. V.: A systematic literature review of literature reviews in software testing. Information & Software Technology 80: 195-216 (2016)
15. Garousi, V., Mäntylä, M.V.: When and what to automate in software testing? A multivocal literature review. Information & Software Technology 76: 92-117 (2016).

16. Gottesheim, W. "Challenges, benefits and best practices of performance focused DevOps," in *Proceedings of the International Workshop on Large-Scale Testing, Austin, TX, USA*, ser. LT '15, ACM, Ed. New York, NY, USA: ACM, 2015, pp. 3–3.
17. Hussaini, S.W., "Strengthening harmonization of development (dev) and operations (ops) silos in it environment through systems approach," in International IEEE Conference on Intelligent Transportation Systems, Barcelona, Spain, IEEE, Ed., Oct 2014, pp. 178–183.
18. Kassab, M., DeFranco, J.F., Laplante, P.A.: Software Testing: The State of the Practice. IEEE Software 34(5): 46-52 (2017).
19. Mahdi Noorian, Ebrahim Bagheri, Weichang Du: Machine Learning-based Software Testing: Towards a Classification Framework. SEKE 2011: 225-229.
20. Moe, N. B., Cruzes, D.S., Dybå, T. and Mikkelsen, E. M.: "Continuous Software Testing in a Globally Distributed Project". ICGSE 2015: 130-134.
21. Rafi, D.M., Moses, K.R.K., Petersen, K., Mäntylä, M.: Benefits and limitations of automated software testing: Systematic literature review and practitioner survey. AST 2012: 36-42
22. Ramler, R., Felderer, M., Leitner, M: A Lightweight Approach for Estimating Probability in Risk-Based Software Testing. RISK 2016: 115-128.
23. Raulamo-Jurvanen, P., Mäntylä, M. and Garousi, V.: Choosing the Right Test Automation Tool: a Grey Literature Review of Practitioner Sources. EASE 2017: 21-30.
24. Taipale, O. and Smolander, K.: "Improving software testing by observing practice". ISESE 2006: 262-271.
25. Whittemore, R, Chase, S. K. and Mandle, C. L.. "Validity in qualitative research." In: Qualitative Health Research 11.4 (July 2001), pp. 117–132
26. Womack, J. P. and Jones, D. T. *Lean thinking: banish waste and create wealth in your corporation*. Simon and Schuster, 2010.
27. Zhang, X., Stafford , T. F. , Dhaliwal, J. S., Gillenson, M. L. and Moeller, G.: "Sources of Conflict between Developers and Testers in Software Development". Inf. & Management, 51(1), 13-26, 2014.