Dimensions of DevOps

Lucy Ellen Lwakatare[1], Pasi Kuvaja[1], Markku Oivo[1]

[1]Department of Information Processing Science, University of Oulu, Finland
{lucy.lwakatare, pasi.kuvaja, markku.oivo}@oulu.fi

Abstract. DevOps has been identified as an important aspect in the continuous deployment paradigm in practitioner communities and academic research circles. However, little has been presented to describe and formalize what it constitutes. The absence of such understanding means that the phenomenon will not be effectively communicated and its impact not understood in those two communities. This study investigates the elements that characterize the DevOps phenomenon using a literature survey and interviews with practitioners actively involved in the DevOps movement. Four main dimensions of DevOps are identified: collaboration, automation, measurement and monitoring. An initial conceptual framework is developed to communicate the phenomenon to practitioners and the scientific community as well as to facilitate input for future research.

Keywords: DevOps · continuous deployment · Agile · software deployment

1 Introduction

Companies offering Internet-based services like Facebook, are now deploying software functionality to customers on a daily basis [1]. This paradigm change towards continuous deployment of software functionality has brought opportunities as well as challenges for most companies [2]. DevOps, a blend of two words, Developers and Operations, is a new phenomenon that helps facilitate this change [2]. It builds a living bridge between development and operations and gives them an opportunity to work and collaborate effectively and seamlessly. Agile methods have improved the performance of software development teams by establishing cross-functional teams and providing closer collaboration with customers [3]. DevOps seeks to extend collaboration of development towards operations, which is responsible for deploying, managing and supporting systems' performance at the customer's site [2]. The continuous deployment paradigm requires software companies to increase communication amongst stakeholders, implement automation and improve agility in designing, delivering and operating software products and services. Erich et al. [4] identified the main concepts of DevOps as: culture, automation, measurement, sharing, services, quality assurance, structures and standards. Bang et al. [5] identified the DevOps perspectives to include: collaboration; automation of build deployment; testing; measurement of the process value, cost and technical metrics; and sharing of knowledge and tools.

Currently, there is lack of common understanding of what DevOps constitutes in academia and in the practitioners' communities. There is a need for research that investigates the DevOps phenomenon and examines how it impacts software development and operations. In this paper, the dimensions of DevOps are identified using relevant academic literature and interviews with practitioners actively involved in the DevOps movement. The main contribution of the study is the definition of the main elements that characterize the DevOps phenomenon and an initial conceptual framework that describes the phenomenon.

2 Research Approach

This study uses a literature review and interviews to investigate the DevOps phenomenon. To identify relevant academic literature, we followed the procedure proposed by Webster and Watson [6].

1. The search term, 'DevOps', was selected and, on 11.11.2014, it was used to retrieve a total of 187 studies from six databases: ACM Digital Library (34), ISI Web of Science (2), Science Direct (10), IEEE Xplore (13), Scopus (28) and Google Scholar (first 100).
2. Relevant studies were selected on the basis of: (a) relevance to the topic, (b) peer reviewed, (c) publication in a scientific journal or in conference proceedings
3. Snowballing was performed in Google Scholar to identify other studies (0).

Interviews. Interviews were conducted with four practitioners working at three companies (Table 1) and actively involved in the DevOps movement as organizers of DevOps Days[1] and DevOps Meetups[2].

Table 1. Description of interviewees.

Company	Offering	Employees	Role of participants
A	ICT products and services	500	**P1**: Lead architect for cloud in Technology Strategy unit
B	ICT R&D services	80	**P2 & P3:** Senior consultants in a DevOps team consisting of developers and operations doing audits for software development process improvements
C	ICT R&D services	63	**P4**: Senior developer, DevOps expert and manager of a technology team

The collected data was imported to Nvivo[3] and analysed following the thematic synthesis approach [7]. The approach involved coding the problems addressed by DevOps, the actions taken (elements) and the impact of the actions taken.

[1] DevOps Days is a technical conference (*http://devopsdays.org/about/*).
[2] DevOps Meetups are face-to-face meetings taking place in different cities around the world.
[3] Qualitative data analysis software (http://www.qsrinternational.com/).

3 Dimensions of DevOps

This section presents the findings from a thorough literature analysis of 22 papers (marked * in the Reference list) and interviews with practitioners. Figure 1 presents an initial conceptual framework that depicts the dimensions that characterize DevOps, the problems that DevOps tries to address and the resulting outcomes.

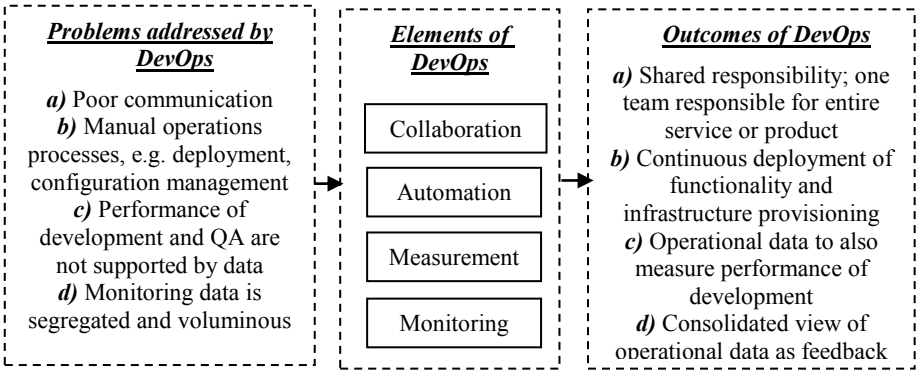

Fig. 1. Conceptual framework characterizing the DevOps phenomenon

3.1 Collaboration

There is consensus in the academic literature and among practitioners that the DevOps phenomenon encompasses a culture of collaboration between the software development organization and the operations organization [2], [8–11]. Other important stakeholders include testers and quality assurance [12]. According to practitioners, it is impossible to effectively transfer information about all releases between two separate organizations in continuous release mode. Other problems include: poor communication between developers and operations and systems designed without complete knowledge and visibility or support for their operational profile [1], [8], [13, 14].

Collaboration is enforced through information sharing, broadening of skillsets and shifting responsibilities between the two teams as well as instilling a sense of shared responsibility [1, 2], [15]. These approaches require changes in people's mind-set as well as changes in the organization's work culture. Collaboration impacts the team structure and the required skillsets of the software developers and operations personnel. The development team may become organized around an entire service taking full responsibility for developing and operating the software functionalities.

3.2 Automation

According to the literature and the practitioners, automation in DevOps is required in operations processes and increased test automation is necessary in the software

development process [9], [15–17]. In order to keep up with the pace of Agile software development and continuous integration (CI) practices, operations processes need to be flexible, repeatable and fast by eliminating manual processes. In complex environments it is difficult and time consuming to manually deploy functionality and manage configurations of software functionality and infrastructure repeatedly and quickly [18, 19]. Additionally, test automation in the CI and customer acceptance phases need to be improved to ensure the quality of the deployed functionality.

One approach that DevOps uses to address the manual process is depicted in the concept of "Infrastructure as a Code" (IaC). The IaC concept is used to describe the idea that almost all actions performed to the infrastructure can be automated [19–21]. It emphasizes developing automation logic for deploying, configuring and upgrading software and infrastructure repeatedly and quickly, particularly in a cloud environment. This is observed through increased adoption of open source configuration tools, such as Chef, in companies (e.g. company A). In the literature, the studies that have contributed to this have focused on defining frameworks that help identify the limitations of the tools and test for their reliability and repeatability [17],[20],[22]. In a cloud environment, automation allows the infrastructure to be provisioned and the functionality to be deployed repeatedly and fast [8], [20].

3.3 Measurement

According to Roche [12], DevOps emphasizes "putting efficiency and process into perspective". This means the ability to measure the development process by incorporating different metrics that will help increase efficiency in product development. Claps [2] further described it as being able to go beyond Quality Assurance (QA) to the system's performance and usage data to seek insights about the quality and usefulness of software functionality.

Measurement in DevOps is achieved by measuring the effort of the software process beyond QA using real time performance and usage data of software functionality in the production environment. The impact is that software development efforts are effectively measured [10].

3.4 Monitoring

Operations personnel monitor systems and the underlying infrastructure to determine the appropriate resource assignment and to detect, report and correct problems that occur during or after system upgrades [13]. According to practitioners, operations personnel use monitoring tools and logs to obtain information regarding a system's health. However, in most cases the logs are voluminous causing developers and operations to spend an extensive amount of time locating the problems, especially when the systems are designed not to expose relevant information [23, 24]. Continuous deployment of functionality further challenges monitoring activities by requiring them to be effective and fast [14], [22]. Another problem addressed by DevOps is that the monitored data is not consolidated and effectively used [8].

DevOps addresses the challenges of effective monitoring by emphasizing collaboration between developers and operations so that the systems are designed to expose relevant information [13], [24, 25]. Additionally, analytics can be used to integrate the system and infrastructure performance data with customer usage behaviour [8]. The information is to be provided as feedback to developers and product management to use for product improvements and customization [2], [8, 9].

4 Conclusion

DevOps is a relatively a new phenomenon that lacks a common understanding and definition in academia and in the practitioners' communities. This study identifies four elements that characterize DevOps: collaboration, automation, measurement and monitoring. A conceptual framework is also presented to describe the phenomenon. This study contributes to previous research by confirming three elements of DevOps and it adds one new element. It also presents a conceptual framework to describe the phenomenon. There is still a need for empirical research that investigates the phenomenon in order to validate and enhance the presented conceptual framework.

References

1. Feitelson, D.G., Frachtenberg, E., Beck, K.L.: Development and Deployment at Facebook. IEEE Internet Computing, 17, 8–17 (2013)*
2. Claps, G.G., Berntsson Svensson, R., Aurum, A.: On the Journey to Continuous Deployment: Technical and Social Challenges Along the Way. Information and Software Technology, 57, 21–31 (2015)*
3. Rodríguez, P., Markkula, J., Oivo, M., Turula, K.: Survey on agile and lean usage in finnish software industry. In: ACM-IEEE International symposium on Empirical software engineering and measurement. p. 139. ACM Press, New York, (2012).
4. Erich, F., Amrit, C., Daneva, M.: Cooperation between Information System Development and Operations. In: 8th International Symposium on Empirical Software Engineering and Measurement, pp. 1–1. ACM Press, New York, (2014)*
5. Bang, S., Chung, S., Choh, Y., Dupuis, M.: A Grounded Theory Analysis of Modern Web Applications: Knowledge, Skills, and Abilities for DevOps. In: 2nd Annual Conference on Research in Information Technology, pp.61-62. ACM , New York (2013) *
6. Webster, J., Watson, R.: Analyzing the Past to Prepare for the Future: Writing a Literature Review. Management Information Systems Quarterly, 26, 13–19 (2002)
7. Cruzes, D.S., Dyba, T.: Recommended Steps for Thematic Synthesis in Software Engineering. In: International Symposium on Empirical Software Engineering and Measurement, pp. 275–284. IEEE Press, New York (2011)
8. Bruneo, D., Fritz, T., Keidar-Barner, S., Leitner, P., Longo, F., Marquezan, C., Metzger, A., Pohl, K., Puliafito, A., Raz, D., Roth, A., Salant, E., Segall, I., Villari, M., Wolfsthal, Y., Woods, C.: CloudWave: Where Adaptive Cloud Management Meets DevOps. In: IEEE Symposium on Computers and Communications, pp.1–6. IEEE Press, New York (2014)*

9. Hosono, S., Shimomura, Y.: Application Lifecycle Kit for Mass Customization on PaaS Platforms. In: 8th IEEE World Congress on Services, pp. 397–398. IEEE Press, New York (2012)*
10. Liu, Y., Li, C., Liu, W.: Integrated Solution for Timely Delivery of Customer Change Requests: A Case Study of Using DevOps Approach. International Journal of U-& E-Service, Science & Technology, 7, 41–50 (2014)*
11. Wettinger, J., Breitenbücher, U., Leymann, F.: DevOpSlang–Bridging the Gap between Development and Operations. In: Villari, M., Zimmermann, W., Lau, K.-K. (eds.) SOCC 2014. pp. 108–122. LNCS, vol. 8745, Springer Berlin Heidelberg (2014)*
12. Roche, J.: Adopting DevOps Practices in Quality Assurance. Communications of the ACM, 56, 38–43. ACM Press, New York, (2013)*
13. Bass, L.: Supporting Operations Personnel through Performance Engineering. In: International Conference on Performance Engineering, pp. 185–186. ACM Press, New York (2013)*
14. Hosono, S., He, J., Liu, X., Li, L., Huang, H., Yoshino, S.: Fast Development Platforms and Methods for Cloud Applications. In: IEEE Asia-Pacific Services Computing Conference, pp. 94–101. IEEE Press, New York (2011)*
15. Cook, N., Milojicic, D., Talwar, V.: Cloud Management. Journal of Internet Services and Applications, 3, 67–75 (2012)*
16. Spinellis, D.: Don't Install Software by Hand. IEEE Software, 29, 86–87 (2012)*
17. Wettinger, J., Andrikopoulos, V., Strauch, S., Leymann, F.: Characterizing and Evaluating Different Deployment Approaches for Cloud Applications. In: IEEE International Conference on Cloud Engineering, pp. 205–214. IEEE Press, New York (2014)*
18. Borgenholt, G., Begnum, K., Engelstad, P.: Audition: A DevOps-oriented Service Optimization and Testing Framework for Cloud Environments. In: Conference of Norsk informatikkonferanse (NIK), pp. 146 – 157, Akademika Publishing, Trondheim (2013)*
19. Harrer, S., Rock, C., Wirtz, G.: Automated and Isolated Tests for Complex Middleware Products: The Case of BPEL Engines. In: IEEE 7th International Conference on Software Testing, Verification and Validation Workshops, pp. 390–398. IEEE, New York (2014)*
20. Hummer, W., Rosenberg, F., Oliveira, F., Eilam, T.: Testing Idempotence for Infrastructure as Code. In: Eyers, D., Schwan, K (eds.) Middleware 2013.pp. 368-388. LNCS, vol. 8275, Springer Berlin Heidelberg (2013)*
21. Wettinger, J., Behrendt, M., Binz, T., Breitenbücher, U., Breiter, G., Leymann, F., Moser, S., Schwertle, I., Spatzier, T.: Integrating Configuration Management with Model-driven Cloud Management based on TOSCA. In: 3rd International Conference on Cloud Computing and Services Science, pp. 437–446, SciTePress (2013)*
22. Zhu, Liming; Xu, Donna; Xu, Sherry; Tran, An Binh; Weber , Ingo; Bass, L.: Challenges in Practicing High Frequency Releases in Cloud Environments. In: 2nd International Workshop on Release Engineering, pp. 21–24, Mountain View, USA (2014)*
23. Xu, X., Zhu, L., Weber, I., Bass, L., Sun, D.: POD-Diagnosis: Error Diagnosis of Sporadic Operations on Cloud Applications. In: 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, pp. 252–263. IEEE Press, New York (2014)*
24. Cukier, D.: DevOps Patterns to Scale Web Applications using Cloud Services. In companion publication for conference on Systems, Programming & Applications: Software for Humanity, pp. 143–152, ACM Press, New York, (2013)*
25. Shang, W.: Bridging the Divide between Software Developers and Operators using Logs. In: 34th International Conference on Software Engineering, pp. 1583–1586. IEEE Press, New York (2012)*