A SURVEY ON DEVOPS IN EDUCATION AND INDUSTRY-SHIFT LEFT PARADIGM CHANGE IN AGILE DEVOPS MODEL

Mrs. V. Shobha Rani*1, Dr. A. Ramesh Babu*2
*1 Research Scholar, Dept. of Computer Science, Chaitanya (Deemed to be University), Telangana, India
*2 Professor, Dept. of Computer Science, Chaitanya (Deemed to be University), Telangana, India

Abstract- **DevOps was seen not just as a change in processes but a major change in the culture of software development. DevOps is a combination of practices and tools that combines software development and information technology operations, is beginning to transform teaching, learning, and research in higher education. First, the existing literature is reviewed to investigate the elements to clarify the concept and provide insight into existing challenges of adopting the DevOps phenomenon in education and shift-left paradigm in DevOps. We performed a Systematic Literature review to explore DevOps and DevOps in education and the importance of shift left testing in DevOps. 15 articles out of 100 were selected, studied, and then formed a definition of DevOps by breaking down the concept into its defining characteristics. we present the main aspects of the educational effort made in the recent years to educate to the concepts and values of the DevOps philosophy and the benefit of the shift-left approach for Faster delivery of software with fewer defects**.

Index Terms-Education Courses, DevOps, Agile software development, Shift-left testing, Industry, Continuous delivery.

INTRODUCTION

Even though the DevOps movement has been discussed for nearly a decade, it lacks a widely accepted definition. By consolidating the most cited definitions of DevOps, we crafted our definition, similar to the one proposed by Dyck et al. [16], which we adopt throughout this article:

DevOps is a collaborative and multidisciplinary effort within an organization to automate continuous delivery of new software versions, while guaranteeing their correctness and reliability.

Desiring to improve their delivery process [33], enterprises are widely adopting DevOps [7, 18,20]. Although in discordance with most of the academic definitions, the software industry also uses the word "DevOps" to describe a well-paid job title [23, 32]. Becoming a DevOps engineer is an attractive opportunity for software professionals. DevOps is also an important phenomenon studied by software engineering researchers and already a mandatory topic in software engineering courses.

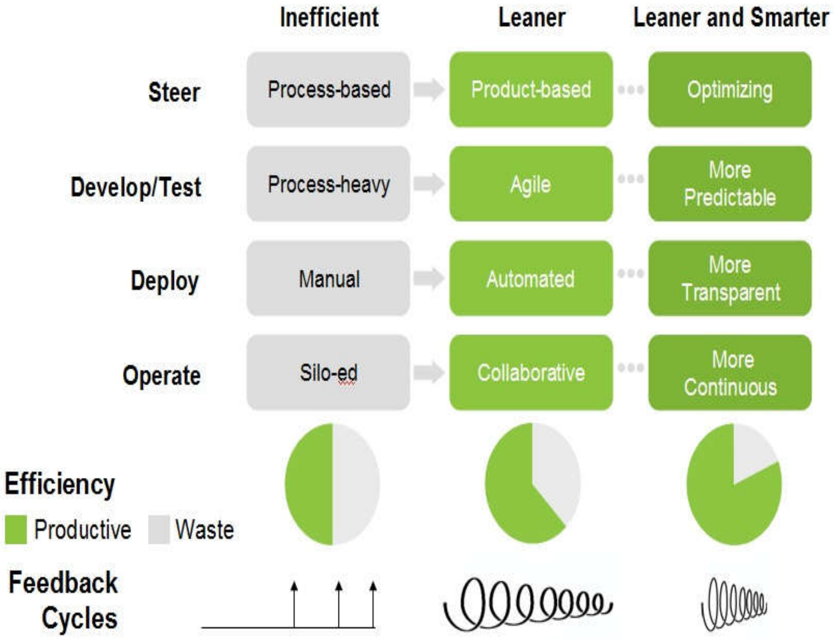
DevOps is an evolution of the agile movement [8]. Agile Software Development advocates small release iterations with customer reviews. It assumes the team can release software frequently in some production-like environment.

Consequently, the transition to production tends to be a stressful process in organizations, containing manual, error-prone activities, and, even, last minute corrections [31]. DevOps proposes a complementary set of agile practices to enable the iterative delivery of software in short cycles effectively. From an organizational perspective, the DevOps movement promotes closer collaboration between developers and operators. The existence of distinct silos for operations and development is still prevalent: Operations staff is responsible for managing software modifications in production and for service levels [27]; development teams, however, are accountable for continuously developing new features to meet business requirements. Each one of these departments has its independent processes, tools, and knowledge bases. The interface between them in the pre-DevOps era was usually a ticket system: Development teams demanded the deployment of new software versions, and the operations staff manually managed those tickets.

In such an arrangement, development teams continuously seek to push new versions into production, while operations staff attempt to block these changes to maintain software stability and other non-functional concerns. Theoretically, this structure provides higher stability for software in production. However, in practice, it also results in long delays between code updates and deployment, as well as ineffective problem-solving processes, in which organizational silos blame each other for production problems.

Conflicts between developers and operators, significant deployment times, and the need for frequent and reliable releases led to inefficient execution of agile processes. In this context, developers and operators began collaborating within enterprises to address this gap. This movement was coined "DevOps" in 2008 [29].

In the *Continuous Delivery* book [31], Humble advocates for an *automated deployment pipeline*,in which any software version committed to the repository must be a productioncandidate version.After passing through stages, such as compilation and automated tests, the software is sent to production by the press of a button. This process is called *Continuous Delivery*. A variant is the *continuous deployment* [30], which automatically sends to production every version that passes through the pipeline. Many authors closely relate DevOps to continuous delivery and deployment [24, 25 26, 28]. Yasar, for example, calls the deployment pipeline a “DevOps platform” [28].Besides automating the delivery process, DevOps initiatives have also focused on using automated runtime monitoring for improving software runtime properties, such as performance, scalability, availability, and resilience [2, 3, 8, 34].



Adoption-Model

LITERATURE SURVEY

Stephen Macdonell [1] illustrated that DevOps was seen not just a change in processes but a major change in culture of software development. Also it presented how gaps surrounding continuous deployment to integrated software be reduced to minimum.

[4] This article aims to clarify the concept and to provide some insight into what is perceived as impediments in the early stages of adopting DevOps. Concluded that DevOps is a multifaceted

concept and its definition still requires attention of the research community. We further conclude that adopting DevOps is not perceived as trivial but it can require overcoming several impediments. Moreover, further research needs to be done on the later stages of the adoption process.

[5] 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.

[6] This paper applied a SLR method and systematically identified a final set of 30 papers. These papers were then reviewed in detail to extract the relevant data and developed catalogues of DevOps concepts, practices, tools, benefits and challenges. These catalogues provide a collective knowledge base of DevOps that can be used by researchers and practitioners to further enhance their understanding and enable effective adoption of DevOps approach in their local context.

[9] This research paper provides an empirical investigation on several tools and challenges encountered during the adoption of continuous practices in Software develops using DevOps. The paper provides several challenges and how several computer science fields can help in dealing with those challenges. Also it was found that relatively few research papers and scholarly articles have been published till date about Devops practices, and the area demands greater extent of research.

[10] Stated DevOps is a conceptual framework, comparable with Agile Software Development. Organizations need to incorporate DevOps principles and practices in their processes. To accomplish this, they will need to restructure themselves.

He has given recommendations for organizations to follow when adopting DevOps. This is merely the first step in creating a comprehensive guide for companies to adopt DevOps. There are other sources which organizations can use in their adoption.

Liamping Chen [11] worked towards adopting continuous deployment which focusses on how software companies can benefit from applying several automated tools for automatic deployment and almost every aspect of SDLC.

Mojtaba Shahin [12] performed the empirical investigation on deployment challenges. It provided the survey results conducted on small and big organizations adopting DevOps regarding the several challenges faced in automatic deployment activity. It was found that lack of better tools and manually driven non- technical activities like bureaucracy also badly impact such practices.

[13] The authors of this paper have experimented for long with novel approaches under different forms. However, DevOps represents a newer and significant challenge. Despite of the fact current educational approaches in academia and industry show some similarities, they are indeed significantly different in terms of attitude of the learners, their expectation, delivery pace and measure of success. Similarities lay more on the perceived hype of the topic, its typical pragmatic and applicative nature, and the minor relevance that education classically reserves to "Operations". While similarities can help in defining a common content for the courses, the differences clearly suggest a completely different nature of the modalities of delivery.

Finally, they have described their vision for the transition to the new curriculum at Innopolis University. In terms of educational innovation, other realities are moving fast and we should not be shy in proposing curricula drastic changes.

[14] Present the main aspects of the educational effort made in the recent years to educate to the concepts and values of the DevOps philosophy. This includes principles, practices, tools and architectures, primarily the Micro service architectural style. Two experiences have been made, one at academic level as a master program course and the other, as an industrial training. Based on those two experiences, we provide a comparative analysis and some proposals in order to develop and improve DevOps education for the future.

[15] Presented the structure of the course and explains what instructional methodologies have been used for course development, such as project based learning that facilitates the students' team based skills both in mastering Agile development process and skills sharing. The paper provides a short summary of the generally used DevOps definitions, concepts, models and tools, specifically focusing on the cloud based DevOps tools for software development, deployment and operation that allows the main DevOps principle of continuous development and continuous improvement which are critical for modern agile data driven companies.

John Ferguson Smart [17] gave the main idea about automated acceptance testing with Screenplay pattern. The idea presented here was to make continuous deployment, configuration of software components possible by making automated acceptance test possible through tests written in view of the actor who interacts rather than interactions with the system.

[19] Gave the idea to shift the test activities left towards the earlier phases of development. This shift ensured that test and QA activities were given the attention they needed. This helped us to

detect defects in both design and code early on, and made it possible to deliver working software to the client by the end of every sprint.

[21] Articulated the trends that are impacting testing. As organizations adopt a continuous delivery approach to developing and delivering software, there is a tendency to focus on feature velocity at the cost of incurring technical debt. This technical debt translates in to defects that ultimately slip in to production. Shifting testing beyond the deployment boundary and continually learn about potential defects from the rich post deployment data is an essential weapon against technical debt.

[22] Understanding the concept of shift left, the need for continuous, shift left testing, define the classification for structuring testing types, analyses, data requirements. We end by providing a conceptual framework for applying shift-left testing for Digital apps.

CONCLUSIONS

In this survey, we have discussed DevOps concepts and challenges presented DevOps adoption model. DevOps is a methodology, comparable with Agile Software Development. Organizations need to absorb DevOps principles and practices in their processes. To achieve this, they will need to re-engineer themselves.

This paper also aids IT professionals by presenting systematically the most relevant concepts, tools, and implications associated with the professional perspectives of researchers, managers, and engineers—including developers and operators.

According to the survey educational institutions require awareness of what DevOps is and what role it has. As they become aware, adoption within higher education is expected to increase significantly, both with new adopters and existing ones. And also discussed the benefit of the shift-left approach for Faster delivery of software with fewer defects.

REFERENCES

[1] Waqar Hussain, Stephen MacDonell on "Emerging Trend for Global Devops", 2017 IEEE 12th International Conference on Global Software Engineering (ICGSE)

[2] Armin Balalaie, Abbas Heydarnoori, and Pooyan Jamshidi. 2016. Microservices architecture enables DevOps: Migration to a cloud-native architecture. *IEEE Softw.* 33, 3 (2016), 42–52. Code: A2.

[3] A. Basiri, N. Behnam, R. de Rooij, L. Hochstein, L. Kosewski, J. Reynolds, and C. Rosenthal. 2016. Chaos engineering.*IEEE Softw.* 33, 3 (2016), 35–41. Code: A76.

[4] Lassenius, Casper; Dingsøyr, Torgeir; Paasivaara, Maria (2015). *[Lecture Notes in Business Information Processing] Agile Processes, in Software Engineering, and Extreme Programming Volume 212 || DevOps: A Definition and Perceived Adoption Impediments. , 10.1007/978-3-319-18612-2(Chapter 14), 166–177.* doi:10.1007/978-3-319-18612-2_14

[5] Lucy Ellen Lwakatare,Pasi Kuvaja, Markku Oivo:"Dimensions of DevOps" in Springer International Publishing Switzerland 2015 C. Lassenius et al. (Eds.): XP 2015, LNBIP 212, pp. 212–217, 2015.

[6] Georges Bou Ghantous, Asif Gill:"DevOps: Concepts, Practices, Tools, Benefits and Challenges" Pacific Asia Conference on Information Systems(PACIS), Summer 7-19-2017, http://aisel.aisnet.org/pacis2017/96

[7] Lianping Chen. 2015. Continuous delivery: Huge benefits, but challenges too. *IEEE Softw.* 32, 2 (2015), 50–54. Code:B15.

[8] Henrik Barbak Christensen. 2016. Teaching DevOps and cloud computing using a cognitive apprenticeship and storytelling approach. In *Proceedings of the ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE'16)*. ACM, 174–179. Code: A47.

[9] Aayush Agarwal; Subhash Gupta; Tanupriya Choudhury**:** "Continuous and Integrated Software Development using DevOps" Published in 2018 International Conference on Advances in Computing and Communication Engineering (ICACCE)

[10] Floris Erich, Chintan Amrit : "Report: DevOps Literature Review" October 2014

[11] Lianping Chen on "Continuous Delivery: Overcoming adoption challenges", The Journal of Systems and Software 128 (2017) 72–86, Science Direct

[12] J. Mojtaba Shahin, Muhammad Ali Babar, Mansooreh Zahedi, Liming Zhu on "Beyond Continuous Delivery: An Empirical Investigation of Continuous Deployment Challenges," In Proceedings of 11th international Symposium on Empirical Software Engineering and Measurement (ESEM), Toronto, Canada

[13] Evgeny Bobrov, Antonio Bucchiarone, Alfredo Capozucca, Nicolas Guelfi:" Teaching DevOps in academia and industry:reflections and vision" Conference paper First Online: 19 January 2020

[14] DevOps and its Philosophy : Education Matters!:Evgeny Bobrov, Antonio Bucchiarone, Alfredo Capozucca, Nicolas Guelfi,Manuel Mazzara, Alexandr Naumchev, Larisa Safina

[15] Yuri Demchenko, Zhiming Zhao, Jayachander Surbiryala, Spiros Koulouzis: "Teaching DevOps and Cloud based Software Engineering in University Curricula" in 15th International Conference on eScience September 2019

[16] Andrej Dyck, Ralf Penners, and Horst Lichter. 2015. Towards definitions for release engineering and DevOps. In *Proceedings of the IEEE/ACM 3rd International Workshop on Release Engineering*. 3–3. Code: B26.

[17] John Ferguson Smart, Antony Marcano, Andy Palmer, Jan Molak on “Automated acceptance testing with Screenplay pattern” IEEE 12th International Conference on Global Software Engineering (ICGSE)

[18] Dror G. Feitelson, Eitan Frachtenberg, and Kent L. Beck. 2013. Development and deployment at Facebook. *IEEE Int. Comput.* 17, 4 (2013), 8–17. Code: B7.

[19] Kristian Bjerke-Gulstuen, Emil Wiik Larsen**:** High Level Test Driven Development – Shift Left**,** May 2015 DOI:10.1007/978-3-319-18612-2_23 Conference: International Conference on Agile Software Development

[20] Jim Gray. 2006. A conversation with Werner Vogels. *ACM Queue* 4, 4 (2006), 14–22. Code: B3.

[21] Kaulgud, Vikrant; Saxena, Amitabh; Podder, Sanjay; Sharma, Vibhu Saujanya; Dinakar, Chethana (2016). [ACM Press the International Workshop - Austin, Texas (2016.05.14-2016.05.22)] Proceedings of the International Workshop on Continuous Software Evolution and Delivery - CSED '16 - Shifting testing beyond the deployment boundary.

[22] Mrs. V. Shobha Rani, Dr. A. Ramesh Babu : “DevOps: Test early Test often-The changing Face of Software Testing”

[23] Waqar Hussain, Tony Clear, and Stephen MacDonell. 2017. Emerging trends for global DevOps: A New Zealand perspective. In *Proceedings of the 12th International Conference on Global Software Engineering (ICGSE’17)*. IEEE Press,21–30. Code: A25.

[24] Mojtaba Shahin, Muhammad Ali Babar, and Liming Zhu. 2016. The intersection of continuous deployment and architecting process: Practitioners’ perspectives. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM’16)*. ACM, 44:1–44:10. Code: A64.

[25] Johannes Wettinger, Vasilios Andrikopoulos, and Frank Leymann. 2015. Automated capturing and systematic usage of DevOps knowledge for cloud applications. In *Proceedings of the IEEE International Conference on Cloud Engineering*.IEEE, 60–65. Code: A61.

[26] JohannesWettinger, Vasilios Andrikopoulos, and Frank Leymann. 2015. Enabling DevOps collaboration and continuous delivery using diverse application environments. In *Proceedings of the On the Move to Meaningful Internet Systems Conferences (OTM’15)*. Springer International Publishing, 348–358. Code: A17.

[27] Eoin Woods. 2016. Operational: The forgotten architectural view. *IEEE Softw.* 33, 3 (2016), 20–23. Code: A82.

[28] Hasan Yasar and Kiriakos Kontostathis. 2016. Where to integrate security practices on DevOps platform. *Int. J. Sec.Softw. Eng.* 7, 4 (2016), 39–50. Code: A58.

[29] Patrick Debois. 2008. Agile Infrastructure Operations. At Agile 2008 Toronto. Retrieved on October 2019 from http://www.jedi.be/presentations/agile-infrastructure-agile-2008.pdf.

[30] Jez Humble. 2010. Continuous Delivery vs Continuous Deployment. Retrieved from: https://continuousdelivery.com/2010/08/continuous-delivery-vs-continuous-deployment/.

[31] Jez Humble and David Farley. 2010. Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation. Addison-Wesley Professional.

[32] Julia Silge. 2017. How Much Do Developers Earn? Find Out with the Stack Overflow Salary Calculator. Retrieved from: https://stackoverflow.blog/2017/09/19/much-developers-earn-find-stack-overflow-salary-calculator/.

[33] Nicole Forsgren Velasquez, Gene Kim, Nigel Kersten, and Jez Humble. 2014. 2014 State of DevOps Report. Retrieved from: https://puppet.com/resources/whitepaper/2014-state-devops-report.

[34] James Roche. 2013. Adopting DevOps practices in quality assurance. *Commun. ACM* 56, 11 (2013), 38–43. Code: A74.