



DevOps automation and Agile methodology

Sikender Mohsienuddin Mohammad

Sr. Associate, KPMG LLP, Department of Information Technology, Wilmington University
419 V street, Apt D, Sacramento, CA 95818

Abstract

In recent times, interest in the collaboration between technology and IT activities which is part of the DevOps phenomenon has been on the rise among, software-engineering, practitioners, and scientists. However, DevOps is also associated with agile as well as continuous software development delivery approaches, given its increasing importance. The study of DevOps follows which are the roots, acceptance, integration, and priorities with agile, lean, and ongoing delivery strategies in software development. DevOps phenomenon was influenced by the development of agile software, based on lean principles. Successful DevOps deployment needs the agile development of software. DevOps is end-to-end automation in developing and delivering software. No one could approach it with a cookbook-like approach, but most developers would benefit from improving connectivity between previously isolated creation and activity silos. Even though agile software development methodologies are more and more common, many organizations have discovered that they have not reached a frequent release rate, largely because of the various departmental features in silos. Organizations have shifted towards DevOps to remove these silos. Companies are increasingly introducing DevOps as digitizations continue. There are three different phases of the agile transition to DevOps: agile, ongoing integration, and constant delivery. Based on a comprehensive case study done in an enterprise with a long background of DevOps, we detect a fundamental disturbance in the soft ability and patterns of communication that software teams are supposed to possess.

Keywords: DevOps, Automation, software engineering, configuration management, agile methodology

I. INTRODUCTION

The distinctions between DevOps and Agile methodologies are very confusing. That is partly because many marketers define these terms loosely which waters down the exact meaning. But it is not just marketing ambiguity that leads to the misunderstanding and confusion of concepts. It also adds to the misunderstanding that DevOps and Agile concentrate on principles rather than practical activities [1]. A software project involves the execution of tasks to achieve added value outcomes. The planning, implementation, and monitoring of these tasks would be the main function that project management plays. The business environment nowadays in terms of software & product design has evolved more rapidly. Project planning, evaluation, management of improvements, and quality control are some of the challenges that differentiate successful and failed projects. Agile methods are widely used and implemented worldwide to deal with these issues [1]. More software developers concentrate on agile development, thereby increasing the productivity of their

projects and meeting the competitive demands of their customers. Agile methodology has been developed to address the problems faced by the conventional model and to offer project teams multiple possibilities during the development process. The methodologies for software development are changing and thus their application development is handled in short times (sprints) and other involving team members are modified through agile and DevOps methodologies [2]. This study will be focused on a qualitative analysis of DevOps and agile methodology responses. The main focus will be to conduct a literature review on what other scholars have discovered when it comes to DevOps automation and Agile methodology and its impact on the current information technology industry. With this in mind, DevOps contributes to a greater understanding of the efficiency of information systems.

II. LITERATURE REVIEW

An analysis was done by Laanti, Salo, and Abrahamsson, Agile and DevOps are buzzing ideas that are commonly used by many organizations, but there is always an often overlap or ambiguity between the two [2]. Only one business model has emerged in software development, but they are implemented beyond software development to the organizational units. These approaches are not the same but are not enemies either. The use of Agile and DevOps in tandem is always the appropriate approach in a team, department, or entity to achieve change. To accept these theories requires being versatile on how they change continuously and to know that no one solution can solve all organizational needs.

I. An Agile methodology

Laanti, Salo, and Abrahamsson state that as software developers are aware of the immediate limitations of heavy-weight processes such as the Waterfall method, they are searching for a way to simplify and make tweaking of software more agile [2]. They have decided to give end-user input more chances to ensure they are on the right track. In the 1990s a large number of lightweight theories, including the famous Scrum and Kanban methods, were developed for software development. In 2001, the Agile Software Development Manifesto codified many of these concepts and put them within the framework of the advancement of agile software. According to Pedrycz, agile methodologies value the following:

- People-especially the teammates, clients, and their interactions — rather than tools and processes [3].
- Immediacy-Functional applications instead of comprehensive documentation.
- Flexibility-To responds to and embraces change, rather than to a predetermined plan.

The agile methodology also weakens the concept of a "finished product," which was the main objective of the Waterfall approach. Agile considers the fact that software development should be iterative and incremental. This means that the customer can execute new features or improve the existing functions on each new software launch. Agile methodologies allow designing software to break down into manageable parts called "user stories." Pedrycz suggests that this underline Agile's value for the consumer, allowing developers to deliver quicker input loops and maintain product compatibility with business needs [3]. Agile also supports adaptive planning, evolving development, early and on-going delivery, and constant improvement to enable developers to respond quickly and flexibly to customer needs, software, and other external factors.

II. The Modern DevOps

Agile was a response to the approaches in waterfall methodologies, but DevOps was not an answer to Agile. Both hypotheses are not the same – because businesses have started to see connections and improved performance and efficiency when used in tandem. Two important areas were known as IT operations (ITOps) and Development Operations (DevOps), which are essential for businesses in the modern era. ITOps' functions include guaranteeing safety, compliance, and reliability while DevOps is in charge of designing and delivering end-user with new products. Even as ITOps provides reliability and safety for all customer needs via the network, DevOps goes hand in hand with flexibility and thorough analysis and communication with the implementation of new software.

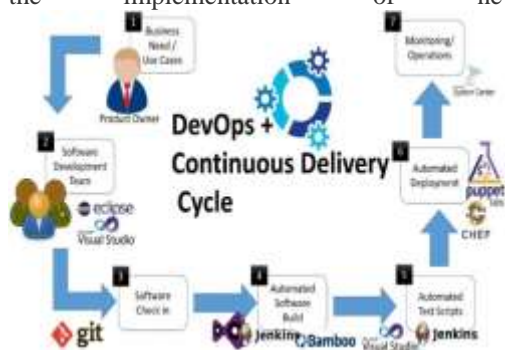


Fig 1: DevOps continuous delivery life cycle
Agile and DevOps are fused [3].

Pedrycz states that in the past decade, firms have tried turning from their original IT team to the specific DevOps group or introducing an agile methodology to their software development projects [3]. Multiple relationships between these two concepts surfaced through these changes in the organization. Agile Teams are focused on automated design, test automation, Continuous Integration (CI). According to Forsgren and Humble DevOps teams frequently need to use these tools, as well as metrics and monitoring systems, configuration management, virtualization, and cloud computing [4]. Agile looked like a whole new paradigm for software developers who were disappointed by the limitations of a waterfall approach. However Agile is not perfect according to Forsgren and Humble. Common disadvantages to Agile planning involve missed deadline, completed elements which, owing to separated scrum or groups, are incompatible with each other and additional features splitting old features – due to the lack of coordination with DevOps and ITOps[4]. All these issues are related to Agile development by one thing: the breakdown in communication.

Forsgren and Humble also suggest that DevOps fills the gap in this case. DevOps is a communications theory, both within itself and across other departments, because developers and operators need to coordinate [4]. DevOps works regularly with ITOps to maintain secure and stable testing environments so it makes sense to turn to other departments such as marketing so customer service by deploying new applications. Advocates of

using both theories in suitable business requirements believe that DevOps can be considered as an extension of Agile methodology. Agile focuses on cross-functional teams, generally consisting of designers, testers, and developers[5]. Drury et al. mention that DevOps takes one step further by introducing an operator who can support the transition between software and implementation. Due to the inherent communications between DevOps and other teams, DevOps can help to automate processes and increase openness for all teams.

content

III. DevOps VS AGILE

Given that the background and discussion dimensions are understood, it is important to explore the comparisons between DevOps and agile methodologies [6]. Agile methodology is a software development tool, which improves the response loop between users and software developers, while DevOps is a composition of cultures, practices, and instruments that divides silos into teams. While DevOps and agile, methodologies have similarities, they are not the same, but some people claim that DevOps is better as compared to agile methodology. It is necessary to get to the nuts and bolts to remove the ambiguity [7]. The contrast at a high level helps to understand the various fields on which both works. **A. Differences**

The differences between agile and DevOps methodologies are as follows:

- Agile methodology is a project management strategy, whereas DevOps works on pipeline optimization.
- Agile concentrates on flexibility in necessities and the advancement of functions, whereas DevOps emphasizes on continuous integration and installation of software developed [8].
- Agile is commonly aligned with systems like Scrum, DaD, LeSS and SAFe, but DevOps does not necessarily apply to specific frameworks.
- Agile emphasizes on operations and DevOps on operational efficiency and automation.

What occurs after software development defines the distinction between the two. In both DevOps and Agile, the software is developed, tested, and deployed. But after these 3 stages, pure agile appears to stop. DevOps, by contrast, includes constant operations [9]. The monitoring and development of software are therefore continuous. In agile, individual people build and test the software and deploy it. DevOps is all responsible for the role of engineering in DevOps; technology is business and software is business. DevOps is more linked to cost reduction and agile is more linked with lean and waste reduction, and terms such as agile accounts for projects and a minimum viable product (MVP) are important. Instead of predictive steps, Agile emphasizes and promotes empiricism (adaptation, openness, and inspection) [10]. When one is not focused on delivering value to the clients, a well-managed and automated production pipeline does no good [11]. The explanation as to why DevOps and Agile are confused is simple: many of the basic principles overlap. The word "continuous delivery" can also be correlated with the two: cooperation, pace, feedback loops. It is important to understand how and why they are distinct and, more specifically, how they can complement one another by gaining a clear [12].

Agile focuses solely on the development and deployment of software while DevOps tends to add IT to the mix. Both are methodologies that are important for software development [13]. The Agile methodology has been in existence for Over 20 years ago while DevOps was recently discovered.

B. Similarities

The concepts of both are centered upon how quickly software can be developed without harming customers nor operations. They both believe in quick software development. Both approaches emphasize speed and consistency [14]. Both are sufficiently malleable and therefore can be integrated into any business models and industries. Both DevOps and Agile are mutually complementary. Agile helps teams to prioritize the versatility, rapidity, and high quality of providing business value to customers [15]. DevOps offers from a technological point of view the underlying organizational culture and infrastructure. The complementary relationship is seen in this light. Well-informed project stakeholders and sprints will mean nothing if the underlying facilities collapse because of the differences in both development and production contexts [16]. The opposite is true, too.

understanding of the two methodologies.

AGILE VERSUS DEVOPS	
AGILE	DEVOPS
A method of software development under which requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their customers	A software engineering culture and practice that aims at unifying software development and software operation
Involves development, testing, integration, and deployment - it does not perform any operations after delivering the product	Involves development, testing, integration, deployment, and operations after deploying the product - it involves continuous monitoring to ensure that the product is performing well
Gets feedback from the customer	Gets feedback from customers as well as from the internal team
Emphasizes more on automation	Uses more open source tools to reduce the overall workload
Minimizes the gap between customer needs and development and testing teams	Minimizes the gap between development, testing and operation teams
Involves small teams	Involves large teams

Fig ii: Differences between Agile and DevOps methodologies [17].

Agile use implementation in large projects

Agile was developed to assist businesses in meeting existing needs in the world of business and a positive and creative working climate. In the literature on how Agile strategies interact with progress, there has been very little documentation of the Organizational Process Improvement Program [18]. Further improvements in technological innovation can be shaped, for example, by corporate business models, concurrent design, multisectoral management, and proactive development [19]. Experts asked on what they felt were the major research issue in their disciplines at the 2010 XP meeting in Trondheim, Norway mentioned that the location of agile as well as large ventures has been the top of the list.

IV. CHALLENGES: AGILE APPROACH

Even though agile technology seems suitable for small teams with direct involvement, multiple barriers exist in large multi-site, multi-customer, and multi-project partnerships to incorporate these practices [20]. As a result, there was a list of transitional challenges and a list of almost forty clear barriers to agile use in large groups. Many of these problems have been identified by the degree of complexity, but others have been identified by the friction between agile and traditional societies;

for example, contradictions between design processes, changes in subsystems that cannot be affected, complex life cycles, and difficulties in using agile heritage frames. Agile design approaches must be tailored to existing procedures in the enterprise [21].

V. AGILE AND DEVOPS CULTURE

Although Agile doesn't always lead to DevOps, the culture changes in an organization are profound. An agile approach will encourage a change in development thinking[22]. Agile thinking encourages tiny, easy-to-use transitions that lead to big improvements over time, rather than thinking of growth as cumbersome [23]. Businesses of all sizes have studied how many units can increase their work in an agile way. Currently, some companies are Agile. DevOps can however bring about its cultural changes, such as improved communications and stable balance with transformation and flexibility [24]. The choice to use both concepts is an active decision, which many analysts think could lead to good decision making and therefore to a better culture of the business.

VI. CONCLUSION

Agile and DevOps are different, even though their correlations cause people to assume that they are the same. This is a disservice to both agile or DevOps. Both help teams to work faster and more efficiently, where quality work is delivered and contented customers. DevOps is a collaboration-building framework for the Technology and Operations team. This allows the consumers to quickly and automatically deploy code for production. It helps to speed up the delivery of products and services through an organization. With these insights in mind, it is clear that both Agile and DevOps strive for a more efficient way to bring the end-users' value— but from various angles. It can be defined as aligning development and IT operation. Agile seeks to improve efficiency for developers and release schedules, while DevOps brings together the operations team to allow continuous implementation and continuous delivery [25]. It has also helped to promote innovative thoughts beyond agile development and multi-functional (DevOps) as well as end-to-end processes that affect software companies and customers. Several organizations have effectively adopted agile techniques to speed up the development of applications. The days of waterfall strategies are long gone. Agile has started to grow slowly into other downstream sectors, including IT and operations, in the technology organization. Teams, simplified processes, improved feedback loops, and driven IT departments become much faster with an innovation that has a profound impact on the whole organization. To build on this performance, DevOps, and Continuous License (CD) have been built to support and improve agility, reactivity, and fast times for marketing over the entire life cycle of software delivery. However, software teams must have a high level of understanding of how to deploy and implement these methodologies. Agile and DevOps are by no means adversaries (or at least they are not intended). Throughout the agile revolution, they are much more allies than enemies. Both Agile and DevOps easily work in the same space either Inclusively or exclusively.

REFERENCES

- [1] Dingsøy, T., Nerur, S., Balijepally, V., and Moe, N., 2012, "A decade of agile methodologies: Towards explaining agile software development", *Journal of Systems and Software*, 85(6), pp. 1213-1221.
- [2] Laanti, M., Salo, O., and Abrahamsson, P., 2011, "Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation", *Information and Software Technology*, 53(3), pp. 276-290.
- [3] Pedrycz, W., 2006, "Quantitative logic-based framework for agile methodologies", *Journal of Systems Architecture*, 52(11), pp. 700-707.
- [4] Forsgren, N., and Humble, J., 2015, "DevOps: Profiles in ITSM Performance and Contributing Factors", *SSRN Electronic Journal*.
- [5] Dzamashvili Fogelström, N., Gorschek, T., Svahnberg, M., and Olsson, P., 2010, "The impact of agile principles on market-driven software product development", *Journal of Software Maintenance and Evolution: Research and Practice*, 22(1), pp. 53-80.
- [6] Drury, M., Conboy, K., and Power, K., 2012, "Obstacles to decision making in Agile software development teams", *Journal of Systems and Software*, 85(6), pp. 1239-1254.
- [7] Forsgren, N., and Humble, J., 2015, "DevOps: Profiles in ITSM Performance and Contributing Factors", *SSRN Electronic Journal*.
- [8] Amorim, A., Mira da Silva, M., Pereira, R., and Gonçalves, M., 2020, "Using agile methodologies for adopting COBIT", *Information Systems*, p. 101496.
- [9] Kamel, M., Bediwi, I., and Al-Rashoud, M., 2010, "Planned Methodologies vs. Agile Methodologies under the Pressure of Dynamic Market", *Journal of King Abdulaziz University-Engineering Sciences*, 21(1), pp. 19-35.
- [10] Spinellis, D., 2016, "Being a DevOps Developer", *IEEE Software*, 33(3), pp. 4-5.
- [11] Saiedian, H., and Dale, R., 2000, "Requirements engineering: making the connection between the software developer and customer", *Information and Software Technology*, 42(6), pp. 419-428.
- [12] Saeeda, H., Arif, F., Mehmood Minhas, N., and Humayun, M., 2015, "Agile Scalability for Large Scale Projects: Lessons Learned", *Journal of Software*, 10(7), pp. 893-903.
- [13] Wettinger, J., Breitenbücher, U., Falkenthal, M., and Leymann, F., 2016, "Collaborative gathering and continuous delivery of DevOps solutions through repositories", *Computer Science - Research and Development*, 32(3-4), pp. 281-290.
- [14] Sidky, A., Arthur, J., and Bohner, S., 2007, "A disciplined approach to adopting agile practices: the agile adoption framework", *Innovations in Systems and Software Engineering*, 3(3), pp. 203-216.
- [15] Petersen, K., and Wohlin, C., 2009, "A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case", *Journal of Systems and Software*, 82(9), pp. 1479-1490.
- [16] Tan, C., and Teo, H., 2007, "Training future software developers to acquire agile development skills", *Communications of the ACM*, 50(12), p. 97.
- [17] Ståhl, D., and Bosch, J., 2014, "Modeling continuous integration practice differences in industry software development", *Journal of Systems and Software*, 87, pp. 48-59.
- [18] Turley, R. T., and Bieman, J. M., 1994, "Identifying essential competencies of software engineers," In *ACM Conference on computer science*, 271-278.
- [19] Ambler, S., and Lines, M., 2012, *Disciplined agile delivery*, IBM Press, Upper Saddle River, N.J.
- [20] Hill, P., 2011, *Practical software project estimation*, McGraw-Hill, New York.
- [21] Hüttermann, M., 2012, "*DevOps for developers*," Berkeley, CA: Apress.
- [22] Gregory, J., and Crispin, L., 2015, "More agile testing," Addison-Wesley, Upper Saddle River, N.J.
- [23] Novak, I., 2012, "Beginning Windows 8 application development," Wiley, Indianapolis, IN.
- [24] Rico, D., Sayani, H., Sone, S., & Safari, an O'Reilly Media Company, 2009, "*The Business Value of Agile Software Methods*," J. Ross Publishing.
- [25] Conboy, K., Coyle, S., Wang, X., and Pikkariainen, M., 2011, "People over Process: Key Challenges in Agile Development", *IEEE Software*, 28(4), pp. 48-57.

