JOURNAL OF SOFTWARE ENGINEERING & INTELLIGENT SYSTEMS
ISSN 2518-8739
30 April 2020, Volume 5, Issue 1, JSEIS, CAOMEI Copyright © 2016-2020
www.jseis.org

DEVOPS APPROACH AND LEAN THINKING IN AGILE SOFTWARE DEVELOPMENT: OPPORTUNITIES, ADVANTAGES, AND CHALLENGES

MAHDI MOUSAEI [1], TAGHI JAVDANI GANDOMANI [2]

[1]*Islamic Azad University, Isfahan, Iran*
[2]*Islamic Azad University, Boroujen, Iran*
Email: moosayimahdis@gmail.com [1], t_javdani@azad.ac.ir [2]

ABSTRACT

The concept of DevOps is a hot topic in recent years. DevOps is a collection of methods for developers and collaborates to deliver software and services rapidly, reliably and with higher quality. The concept of 'lean' defines a kind of the operation in the Toyota Production System. Continuous Integration (CI) and Continuous Delivery (CD) are one of the important practices in the software development. CI is one of the agile software development practice that involves automation and frequent integration of the software changes. DevOps and CD are another subset of the agile. DevOps could be extended with agile methods and by using patterns and practices it helps to improve collaboration between development and operation teams. Both development models of agile and lean are similar to in terms of goals by focusing on the customers and responding to their needs in a rapid method. Agile software development focuses on the software development function, but Lean Thinking has a very explicit focus on the end-to-end process. In this review article, the principle and practices of DevOps, reasons for its usage, challenges of DevOps, the relationship of DevOps with agile are studied.

OPEN ACCESS

Academic Editor: Farrukh Zeeshan, COMSATS, Pakistan.
Published: 30 April, 2020
Funding: Authors have no funds.
Article Type: Research Article

Keywords: agile development; DevOps; continuous deployment; lean software development; continuous integration;

1. INTRODUCTION

In many IT departments, the DevOps can deliver a higher quality of the software [1]. DevOps is a collection of methods in which developers and operations team collaborates to deliver software and services rapidly, reliably and with higher quality. The term of DevOps was generalized through a series of "DevOpsdays" starting in 2009 in Belgium. Also, DevOps described as a software development method that combines quality assurance mechanisms with IT operations within software engineering practices (see Figure. 1)[2].

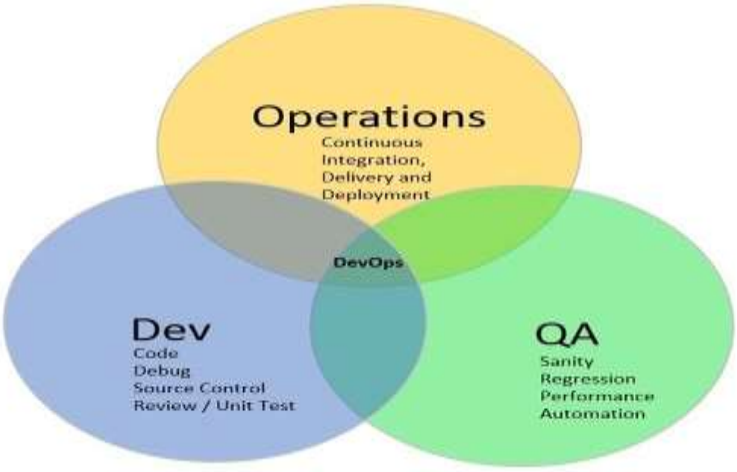

Figure. 1 DevOps concept

Corresponding Author: Mahdi Mousaei
Email Address: moosayimahdis@gmail.com

DevOps is created on the ideas from agile development and supports rapid development and deployment cycles [2]. Agile methods and DevOps are one of the hot topics in the field of software engineering (see Figure. 2).

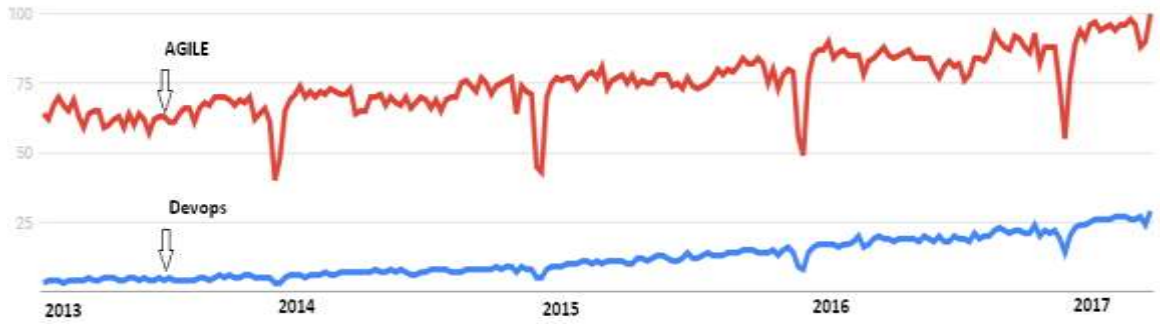

Figure. 2 Relative interest over time on agile methods and DevOps based on searches in Google Trends

Figure. 2 shows agile methods and DevOps trends in the last five years. It is based on Internet searches[1]and indicates relative interest among developers on topics. Study about agile development and DevOps is a challenging topic in software engineering [3].

2. PRINCIPLE AND PRACTICES OF DEVOPS

The main DevOps principles and practices have categorized in Table 1 [4, 5].

Table. 1 Principle and Practices of DevOps [4, 5]

DevOps Principles	DevOps Practices
Iterative	Self-service configuration
Continuous	Automated provisioning
Automated	Continuous build
Self-service	Continuous integration
Collaborative	Continuous delivery
Holistic thinking	Automated release management
Incremental	Incremental testing

3. ADVANTAGES OF SHIFTING TO THE DEVOPS APPROACH

Companies that used of practices DevOps have described many advantages, including more speed up time to market, building the valuable product, improved productivity and efficiency, more reliable releases, improve product quality, and customer satisfaction [6]. DevOps cause to Optimization in companies. This Optimization allows companies to achieve their goals of increasing income and decrease costs and making a good team that explained as follows. [7].

3.1 Increasing income

Increasing income depends on releasing the product or service to market faster, which requires ever-faster, and reliable software build and deployment methods. With combined smaller teams into larger teams with common purposes, DevOps facilitates a broader understanding of the system. By working together, designing process and problems solvation could be done rapidly [7].

3.2 Decrease costs

Decrease costs by bringing down the rework need building quality in the process. When the members of the teams have the capacity to work on improvements instead of suspensions and rework can increase the quality of the project [7].

3.3 Good team

Create a good team can enable them to take honour in their work. Working in a dynamic team enhance motivation and efficiency in the project. Teams by using a system thinking technique can help for continuous improvement of the skills. [7], moreover, with using agile principles can create a good team.

[1] Searches in Google analyzed by Google Trends.

4. SOFTWARE DEVELOPMENT PRACTICE

Recent trends show that required an approach for continuous integration (CI) of the software. Emphasis on DevOps recognizes that the integration between software development and operational deployment needs to be continuous [8]. Rather than focusing on agile methods can use continuous concepts in software development with lean approach [9]. In recent years lean software development has been very much considered [10, 11].

4.1 Continuous integration

Continuous integration (CI) is a hot topic in software development [12]. It is a software development practice where software is integrated continuously during development. Many teams discover that this approach leads have increased more rapidly to developing integrated software [13]. Continuous integration is an agile software development practice, involving automation and frequent integration of software changes [14]. CI requires a link between development and operations team and it very matched to the DevOps [15].

4.2 Continuous delivery

Continuous delivery (CD) causes valid software is built to the customers [16]. The CD as a software engineering approach where the development team releases the software in short cycles [17]. Continuous Delivery should be considered from both an engineering perspective and a business perspective [18]. DevOps and Continuous Delivery is another subset of agile. The CD is a group of practices in software development that are created to improve the processes of software delivery and reliable software releases [19]. The CD is built on the agile principles to resolve some of the agile bugs as like communication, processes, and aspects of tools.

5. AGILE TOOLS FOR DEVOPS

There are many tools that can be used high-efficiency with DevOps. Teams require tools such as configuration tools, maintaining tools and how to manage workflows in projects. In this review introduced a few agile tools. DevOps does not have any formal list of which tools to use, but the following tools are often described by DevOps and agile developers in books and text written [4].

5.1 Choosing the suitable tools

RANCID is an abbreviation of "Really Awesome New Cisco config Differ". RANCID helps to control and manage all configuration. RANCID can help large and complex software development projects [4]. When it correctly installed, it provides an easy way to access and manage all device configuration in the system for the user. Also, Puppet is similar to the configuration tool like RANCID [20]. A Puppet is a powerful tool for DevOps, but there are many add-ons that can be added to Puppet to increase its functionality. There are many tools available that can perform the needs of any DevOps team [4].

5.2 Managing Workflows

Managing workflows is one of the main parts of the DevOps that is in the DevOps process to be all the time evolving. The approach of SCRUM and Kanban are the two important methods in software development for workflow management. Scrum is well known to any developer, and while it is not meant specifically for DevOps particularly it is an integrated part of software development. Also, Scrum is very important to the DevOps development process. Scrum fits into DevOps mind really well because it is designed as a way for teams to work together to successfully develop software. Scrum focuses heavily on the human factor that is important in DevOps. Kanban similar to scrum has imagined and improved to workflow and bottlenecks [4]. Also, right now Kanban is the more famous tool between software developers [21].

6. CHALLENGES OF DEVOPS

DevOps can control the gap between market requirements and traditional software engineering approaches by seeking to decrease the gap between Development and IT Operations. DevOps is a collection of best practices and methods by using agile methods that focus on better collaboration between the two groups. One of the important successful adoption of DevOps and Continuous Delivery are quality, automation, collaboration, and governance/process. With these fundamental elements can be created agility across the end-to-end application lifecycle. DevOps ensures that will be Resolved the current gap between development and IT operations, with using best practices/processes on different layers like Quality, Automation, Collaboration and Governance (see Figure. 3) [22].

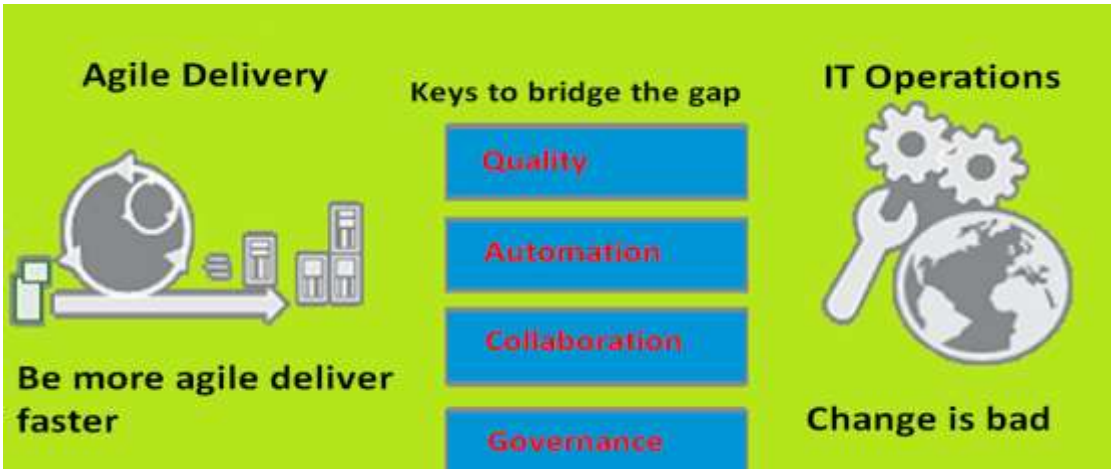

Figure. 3 DevOps delivery model [22]

Quality layer cause to deliver the faster method and make the application more stable and produce releases with higher quality purpose. Automation layer cause improves delivery speed, throughput/productivity, and repeatability. Collaboration layer cause to better communication between different teams within the same project via different communication approaches/tools to minimize risk, rework cost and maximize value towards the customer. Governance layer is the main responsibility to control how these layers work together to ensure better achievement of the global objective of introducing the DevOps delivery model [22]. Joonas Hamunen examined the challenges of the DevOps in four categories [23]. This challenge Contains 16 sub challenge that in continue will be described.

6.1 Lack of knowledge in DevOps

The concept of DevOps is not well known, this concept is not yet enough to grow. The lack of knowledge inside an organization creates an impasse. The summary of challenges about loss of knowledge in DevOps is shown in Table 2.

Table. 2 Challenges Lack of the knowledge in DevOps [23]

Maturity of concept	One of the main challenges is the real concept of DevOps. The lack of definition of the concept of DevOps Cause confusion in software projects. These problems cause Rejection of DevOps in organizations.
Sensitivity to some of the words	DevOps is currently a hot word and ambiguity. Some believe that should be dropped the umbrella term 'DevOps' and very talking about reducing the time and introducing automation.
Lack of the knowledge	DevOps is a new concept for many people. The lack of knowledge about DevOps is one of the important challenges for many researchers. The concept of DevOps is not confined to the components such as Continuous Delivery or Continuous Integration.

6.2 Lack of support for DevOps

One of the main important challenges is lack of support for DevOps. The lack of support has different levels. A summary challenges Lack of support for DevOps shown in Table 3.

Table. 3 Challenges Lack of support for DevOps [23]

Lack of management support	Lack of management support is one important another challenge in DevOps. The lack of actual support from management is one of issue decision to accept the DevOps.
Lack of team-level support	The change of working methods is the deepest change in DevOps.
Lack of trust	Lack of trust is one of the challenges explicit at the very top level in the organizations.

6.3 Implementing DevOps technology

The key success in DevOps is the creation of an automated continuous delivery pipeline. This new method of delivering software has a deep impact on the processes in development, quality assurance, and operations. Implementing this technology has challenges in development. A summary of these challenges in DevOps technology shown in Table 4.

Table. 4 Challenges of Implementing DevOps technology [23]

Automated testing	Usually, the test is done in a final step. But in DevOps automation tests it should be concurrently as a part of the development process.
Automation challenges	The other challenge is dependent on tools many of them are fully new and they don't support all needed environments.
Type of application	The most common problem is that the application architecture is not appropriate for the virtualization.
Finding the right scope for monitoring	Using of DevOps and new tools, for example, the cloud-based New Relic, it's suitable and easier to monitor different services in an integrated vision.

6.4 Adapting organizational processes to DevOps

One of the other objectives of DevOps is increased speed and flexibility in the delivery of software. If this objective fully not satisfactory, might the organizational processes don't complete. A summary of challenges, lack of adapting organizational processes of DevOps is shown in Table 5.

Table. 5 Challenges of Adapting organizational processes of DevOps [23]

Starting with the correct scope	This challenge related to the concept of starting with the correct domain with attention to using the DevOps.
The mode of Software Development	There is a very common challenge related to Scrum. The main important challenge in organizations are developing the software in sprints without the goal of releasing the software.
Change Management Processes	The next main challenge with use DevOps continuous delivery is how it's coordinated in the organizations with current changes and release management processes.
Adopting new metrics	The next challenge is detection the balance between improving the score of the metrics and keeping up a fast speed in development. Uptime and the value of tasks completed in traditional metrics are still acceptable. Lead time and code quality in the "DevOps" metrics can be measured with new types of tools.
Team challenges	The location and time with other teams and product teams have related challenges. Researchers believe that having development and operation team in the isolated locations is the biggest challenge for DevOps.

7. RELATIONSHIP BETWEEN THE CONCEPT OF DEVOPS AND AGILE

DevOps can provide a practical development of the agile activities. DevOps emphasis more on the communication and collaboration between developers and operators rather than tools and processes, it can extend agile principles to software delivery pipeline [24]. Agile and DevOps are similar to each other because agile software development represents a change in thinking and practice (that should lead to organizational change), but DevOps more emphasis on implementing organizational change to achieve project goals. DevOps could be developed with agile and combined with patterns and practices to improve collaboration between development and operation teams [25]. DevOps and Continuous Delivery (CD) is another subdivision of agile which the team holds its software ready to release at all times during development. Both DevOps and agile is synchronized to release the software products [26]. While Agile and DevOps are linked, but they are different in some key aspects. Agile is just a change in the thinking that encourages organizations to bring a change with continuous feedback from customers. On the other hand, DevOps is the real implementation of organizational cultural change [27]. Some of the parameters of relationship DevOps and Agile are shown in Table 6.

Table. 6 The summary of parameters of relationship DevOps and Agile

Development agile with DevOps	DevOps can provide a practical development of the agile activities. DevOps emphasis more on the communication and collaboration between developers and operation teams rather than tools and processes. Also, it can extend agile principles to software delivery pipeline [24].
DevOps and Agile web application	Bayser et al [28], they concluded that in DevOps, there is a special relation between agile web application development and delivery.
Agile as an enabler for DevOps	Hosono [29] believes that agile methods can considered as enablers to adopt DevOps thinking.
Agile supports DevOps	Agile can support DevOps by encouraging collaboration among team members, automation of build, deployment, and test, measurement and metrics of cost, value and processes, knowledge sharing and tools [30].

DevOps against Agile	Terhi et al. [31] believe that Agile methods for continuous integration and deployment have shared properties with DevOps, while DevOps itself cannot meet all principles proposed in the agile manifesto.

8. SOFTWARE DEVELOPMENT LIFECYCLE

A software development life cycle (SDLC) is an important phase for developers, such as planning, analysis, design, and implementation a software product [25]. The lifecycle defines a methodology for improving the quality of software and the overall development process [32]. The Waterfall model is one of the oldest methods, while agile software development is a modern development. Also, some believe DevOps is the third generation software development method that is the continuation of the second generation Agile methods [23].

8.1 Lean software development

At first the term of "lean" described the mode of operation in the Toyota Production System (TPS) originally called "just-in-time production"[33]. Lean Software Development shows that software professionals how to achieve high quality, speed, and business value by adapting the seven of "lean" principles that have already revolutionized manufacturing (see Figure. 4) [34].

Figure. 4 Seven principles of Lean Software Development

Lean software development emphasizes developing powerful, change-tolerant design so that it can be adapted to the kinds of changes [34]. In this section introduced a number of keys lean concepts that be observed in many software engineering practices [15]. Also, in this review article, introduced a number of keys lean concepts that can be examined in the software engineering practices (see Table 7 to a summary of terms) [35].

Table. 7 Lean thinking terminology and examples in software development [35]

Lean term	**Example in software engineering**
Andon	Traffic light connected to continuous integration server and it is a tool to achieve jidoka
ChakuChaku	Automated delivery pipeline
Genchienbutsu	“Daily stand-up meeting
Hanedashi	Automatic ejection, for instance, automatic testing of newly checked in code
Heijunka	Workload leveling of features with a Kanban
Jidoka	Tools and techniques to detect faults so that they can be corrected as soon as possible, e.g., continuous integration
Kaikaku	Reimplementation, architectural overhaul, transition from waterfall to agile
Kanban	Kanban board to let developers ‘pull’ stories to implement once they have the capacity to do so
Kaizen	Sprint retrospective meetings, refactoring

Obeya	Scrum board
Poka Yoke	Fool proofing mechanisms to prevent mistakes and defects early, e.g., syntax highlighting, unit tests, static source code checkers such as splint"
Single-minute Exchange of Dies (SMED)	"Automatic deployment of a new software version with a single press of a button"

8.1.1 Value and waste

A fundamental focus on lean thinking is to shorten the time between a customer order and the delivery of that order; any activities that don't add 'value' so are considered 'waste' [36]. In lean thinking, values are defined as the starting point by the customer [37]. TPS (Toyota Production System), identified the seven types of waste that include [36]:

1. Waste of overproduction
2. Waste of time on available (waiting)
3. Waste of transportation
4. Waste of processing itself
5. Waste of stock on available (inventory)
6. Waste of movement
7. Waste of making defective products

8.1.2 Flow and batch size

Flow is another concept within Lean Thinking [38]. Flow can be compared with 'batch- and-queue' thinking, in which actions are done in batches of products, after which they are queued for the next processing stage. flow mention to set of product feature are included identified, designed, implemented, integrated, tested, and deployed. Some of the traditional software development environments are working with the principles of batch-and-queue. May software development function in the planning and deployment of features is done in batches, and it isn't in a continuous flowing movement. In addition, the software industry is adopting kanban as an alternative approach for scheduling work, or as a source determinate Scrum method. Kanban can help with the daily workload [35]. Agile software development is focused on the software development function, but Lean Thinking has a very explicit focus on the end-to-end process: from customer to delivery (see Figure. 5) [39], [40].

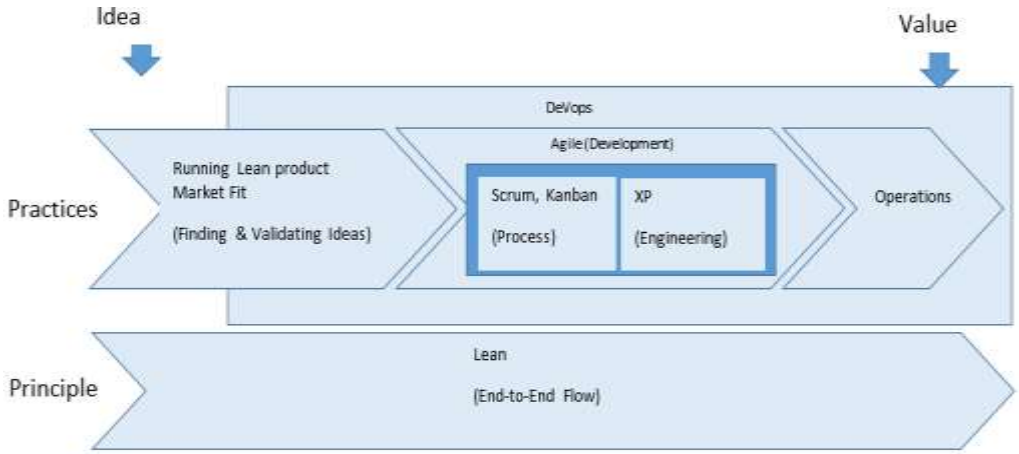

Figure. 5 The relationship between Agile, DevOps and lean approach together [39]

A comparison of agile and lean software development, the end-to-end focus is the key difference between agile and lean. Also lean has adopted many practices known in the agile context, while lean emphasize the importance of using practices that are relevant to the end-to-end flow. [40].

8.1.3 Kaizen and continuous development

The "lean thinking" is a continuous process of betterment. Betterment can be accessed through radical steps (kaikaku) in the inception of an evolution initiative, followed by incremental improvements (kaizen). For example, the decision to use an agile methodology such as Scrum instead traditional methods, the plan-driven methods is an example of 'kaikaku,' whiles the review and retrospective meetings at the end of a sprint is a type of 'kaizen.' Daily Scrum meetings are similar to the daily build summary meetings found within the Toyota Product Development System (TPDS) [35].

9. LEAN AND AGILE APPROACH

Both development model agile and lean seems similar in their aim focusing on the customers and responding to their needs in a rapid method. However, in some aspects are different (see table 8) [40, 41, and 42].

Table. 8 A comparison of Goals in lean and agile [41]

Aspect	Lean	Agile
Customer	Create a product with high quality	Creating an efficient and valuable product that will meet their customer needs
Quick Development	Creating valuable product in less time	The release of working software
Approach	Create a holistic approach to make an organization or process more effective and efficient.	Include techniques suitable for the software development industry.

With Lean Development can be expanded the theoretical foundations of agile software development by using lean principles to software development [43]. In recent years, most companies are using lean and agile methods to improve their products [37]. One of the advantages of combination agile and lean is that the lean can help the agile method to scale up because the lean can be applied to projects that the agile is not suitable. One of the reasons for combining agile and lean is that the agile software processes can be improved by using lean practices [44]. "Scrumban" is a term created in real-world as an example of the combination of agile and lean practices [45, 46]. It is a software production model based on Scrum and Kanban. Table 9 indicates the summary of different combinations of agile and lean in software development [44].

Table. 9 Different combinations of agile and lean in software development [44]

Difference between Agile and Lean	Combination Type
Agile and lean are different because Lean is thinking tool, but agile is a prescriptive method.	Use the lean processes to guide the development and adaptation of agile methods.
Agile and lean have different scopes and focus, but are alike in levels.	Lean is a Top-down implementation to create an environment while agile is bottom-up.
	For scale up in agile use lean techniques.
	By using lean techniques will be improved agile software development processes.
	Use the agile methods to support the lean software development processes.

10. THE CHALLENGES THAT DEVOPS TRIES TO OVERCOME

The organizational and cultural obstacle is one of the most conclusive challenges in DevOps [47]. Several researchers believe the misalignment on purpose and motivation in DevOps is difficult. Agile methodologies have done very much to increase collaboration and transparency within the development group and quality assurance. In Table 10 shows the summary of the problems with agile development and solutions with DevOps [47].

Table. 10 Summary of the problems with agile development and solutions with DevOps [47]

Problem with agile development	DevOps solution
Release the new product to the customer is often delayed.	DevOps tools are used to test and release new features.
Incompatibility of software components with each other.	Make independent components by using Open interfaces and test automation
Unclear of a product quality before the release of product.	By using DevOps tools and practices can increase the automating quality assurance.
Losing time and budget.	The tools and processes of DevOps increase the transparency and predictability of the project.
Non-cooperation of developer teams and IT operations.	Cooperates the developer teams and IT operation together and Their goals are integrated.

11. CONCLUSION

In this research, discussed the concept DevOps approach and lean thinking in agile software development and after its opportunities, benefits and challenges are also examined. Now concept DevOps is helping to deliver

higher quality services. Companies that use of practices and principles of DevOps have described many advantages. DevOps can provide a practical development of the agile activities. DevOps could be extended with the agile and mix together of patterns and practices to improve collaboration between development and operations team. Lean Software Development shows that the software professionals how to achieve high quality, speed, and business value by adapting the seven of "lean" principles. The Lean and agile approach has similarities and differences that discussed in this research.

REFERENCES

1. Spinellis, D. *Being a DevOps developer*. IEEE Software, (2016), 33(3), 4-5.
2. Olszewska, M., & Waldén, M, *DevOps meets formal modelling in high-criticality complex systems*. In Proceedings of the 1st International Workshop on Quality-Aware DevOps (2015), (pp. 7-12). ACM.
3. Kersten, M, A Cambrian *Explosion of DevOps Tools*. IEEE Software, (2018), 35(2), 14-17.
4. Kristinsson, R. *Software Development with DevOps*. the thesis, University of Applied Sciences, (2015).
5. Chris Haddad .2014. Available from: https://dzone.com/articles/DevOps-DevOps-principles [Accessed 14 March 2017].
6. Chen, L. *Continuous delivery: Huge benefits, but challenges too*. IEEE Software, (2015), 32(2), 50-54.
7. DeGrandis, D. "*DevOps: A Software Revolution in the Making,* (2011).
8. P. Debois. *DevOps days* ghent, http://www.DevOpsdays.org/events/2009-ghent/, (2009).
9. Reinertsen, D. G. *Principles of Product Development Flow*. Redondo Beach: Celeritas Publishing, (2009), (Vol. 1).
10. Kobus, J. *Demystifying lean IT: conceptualization and definition*, (2016), MKWI 2016 proceedings.
11. Gandomani, T. J., & Nafchi, M. Z. *An empirically-developed framework for Agile transition and adoption: A Grounded Theory approach*. Journal of Systems and Software, (2015), 107, 204-219.
12. Engblom, J. *Continuous Integration for Embedded Systems using Simulation*. In Embedded World 2015 Congress, (2015).
13. Fowler, M., & Foemmel, M. H *Continuous integration. Thought-Works* http://www. thought works. com/Continuous Integration. pdf, (2006),122.
14. Ståhl, D., & Bosch, J. *Automated software integration flows in industry: a multiple-case study*. In Companion Proceedings of the 36th International Conference on Software Engineering, (2014, May), (pp. 54-63). ACM.
15. Fitzgerald, B., & Stol, K. J. *Continuous software engineering and beyond: trends and challenges*. In Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering, (2014, June), (pp. 1-9). ACM.
16. Amaradri, A. S., & Nutalapati, S. B. *Continuous Integration, Deployment and Testing in DevOps Environment*. The MSc thesis, Blekinge Institute of Technology, Karlskrona, Sweden, (2016).
17. Neely, S., & Stolt, S. *Continuous delivery? easy! just change everything (well, maybe it is not that easy)*. In Agile Conference (AGILE), 2013 (pp. 121-128). IEEE.
18. Erich, F., Amrit, C., & Daneva, M. Report: *DevOps literature review*. University of Twente, Tech. Rep, (2014).
19. Mohamed, S. I. *Software Release Management Evolution-Comparative Analysis across Agile and DevOps Continuous Delivery*. International Journal of Advanced Engineering Research and Science, (2016), 3(6).
20. Thurman, J. 5 *Unsung Tools of DevOps*. " O'Reilly Media, Inc.", (2013).
21. Hammarberg, M., &Sunden, J. *Kanban in action*. Manning Publications Co, (2014).
22. Mohamed, S. I. *DevOps shifting software engineering strategy Value based perspective*. International Journal of Computer Engineering, (2015), 17(2), 51-57.
23. Hamunen, J. *Challenges in adopting a DevOps approach to software development and operations*. the MSc thesis, Aalto University. Espoo, Finland, (2016).
24. Jabbari, R., bin Ali, N., Petersen, K., &Tanveer, B.*What is DevOps?:A Systematic Mapping Study on Definitions and Practices*. In Proceedings of the Scientific Workshop Proceedings of XP, (2016), (p. 12). ACM.
25. Perera, P., Bandara, M., &Perera, I. *Evaluating the impact of DevOps practice in Sri Lankan software development organizations*. In Advances in ICT for Emerging Regions (ICTer), Sixteenth International Conference, (2016), (pp. 281-287). IEEE.
26. Mohamed, S. I. *Software Release Management Evolution-Comparative Analysis across Agile and DevOps Continuous Delivery*. International Journal of Advanced Engineering Research and Science, (2016), 3(6).
27. Yvonne Burns .2017. "*How Does Agile Methodology and DevOps Framework Apply to Delivering Digital Services?"*. Available from: http://www.mastechdigital.com/blog/agile-methodology-DevOps-framework-apply-delivering-digital-services. [Accessed 15 May 2017].

28. De Bayser, M., Azevedo, L. G., & Cerqueira, R. Researchops: *The case for DevOps in scientific applications*. In Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on, (2015), (pp. 1398-1404). IEEE.
29. Hosono, S. *A DevOps framework to shorten delivery time for cloud applications*. International Journal of Computational Science and Engineering, (2012), 7(4), 329-344.
30. Bang, S. K., Chung, S., Choh, Y., & Dupuis, M. (2013, *A grounded theory analysis of modern web applications: knowledge, skills, and abilities for DevOps*. In Proceedings of the 2nd annual conference on Research in information technology, (2013),(pp. 61-62). ACM.
31. Kilamo, T., Leppänen, M., &Mikkonen, T. (2015, September). *The social developer: Now, then, and tomorrow.*
32. Techopedia Inc.2017.Available from:https:// www.techopedia.com/definition /22193/ *software - development -life-cycle-sdlc* [Accessed 14 March 2017].
33. Krafcik, J., *Triumph of the lean production system*. MIT Sloan Manage, 1988, Rev. 30 (1), 41–52.
34. Mary, P., & Tom, P. *Lean software development: an agile toolkit*. Addison Wesley. In Proceedings of the 7th International Workshop on Social Software Engineering, (2003). (pp. 41-48). A
35. Fitzgerald, B., & Stol, K. J. *Continuous software engineering: A roadmap and agenda. Journal of Systems and Software*, (2017), 123, 176-189.
36. Ohno, T., *Toyota Production System: Beyond Large-Scale Production*. CRC Press, (1988).
37. Shahzeydi, M., & Gandomani, T. J. *Adding lean principles to agile software development: A case study* report. International Journal of Software Engineering and Technology, (2016), 2(1).
38. Womack, J., Jones, D.T., Lean Thinking: *Banish Waste and Create Wealth in Your Corporation.* Productivity Press, (2003).
39. Matthias marschall.2012. Available from: http://www.agileweboperations.com/lean-agile-DevOps-related [Accessed 19 March 2017].
40. Petersen, K., *Is lean agile and agile lean? a comparison between two software development paradigms. Modern Software Engineering Concepts and Practices*: Advanced Approaches. IGI Global, (2011)
41. Petersen, K. *Implementing lean and agile software development in industry*, the Doctoral Dissertation, School of Computing, Blekinge Institute of Technology, Sweden, (2010)
42. Rupesh Lochan, 2011. *Lean or Agile? A Comparison of Approach* Available from: https://www. processexcellencenetwork.com/lean-six-sigma-business-transformation /articles/using-lean-in-agile-software-development-a- comparison [Accessed 19 March 2017]**.**
43. Poppendieck, M., &Poppendieck, T. *Lean Software Development: An Agile Toolkit: An Agile Toolkit*. Addison-Wesley.CM, (2003).
44. Wang, X. *The combination of agile and lean in software development: An experience report analysis*. In Agile Conference (AGILE), (2011), (pp. 1-9). IEEE.
45. Ladas, C. Scrumban. *Lean Software Engineering-Essays on the Continuous Delivery of High Quality Information System*, (2008).
46. Ladas, C. *Scrumban-essays on kanban systems for lean software development*. Lulu. Com, (2009)
47. Hamunen, J. *Challenges in adopting a DevOps approach to software development and operations*. the MSc thesis, Aalto University, Finland, (2016).

AUTHORS PROFILE

Mahdi Mousaei received the M.Sc. in Software Engineering from University of Isfahan (Khorasgan), Iran in 2018. His research interests in software engineering are Agile software development, software testing, software quality, requirement engineering, and risk management in agile software development.

Taghi Javdani Gandomani is a PhD candidate in Software Engineering at the University Putra Malaysia. His research interests in software engineering are Agile software development, Software Process Improvement, Development Methodologies and Empirical studies. He is also lecturer in Islamic Azad University, Iran and has more than 14 years industry experience in software development.