Innovation Agile Methodology towards DevOps

Poojan S Patel

Department of Computer engineering, Sardar Vallabhbhai Patel Institute of Technology (SVIT), Vasad, INDIA.

---***---

Abstract - *Interest in the DevOps concept, which involves collaborating across technology and IT activities, has grown rapidly in recent years among software engineers, users, and experts. Given its growing value, DevOps is also associated with agile and continuous software development delivery methodologies. The analysis of DevOps focuses on the roots, acceptance, integration, and priorities of agile, lean, and continuous software delivery methodologies. The development of agile software based on lean principles encouraged the DevOps concept. Agile software development is required for successful DevOps implementation. DevOps is the automation of the entire software development and delivery process. As digitization continues, corporations are turning toward DevOps, and corporations are deploying DevOps.*

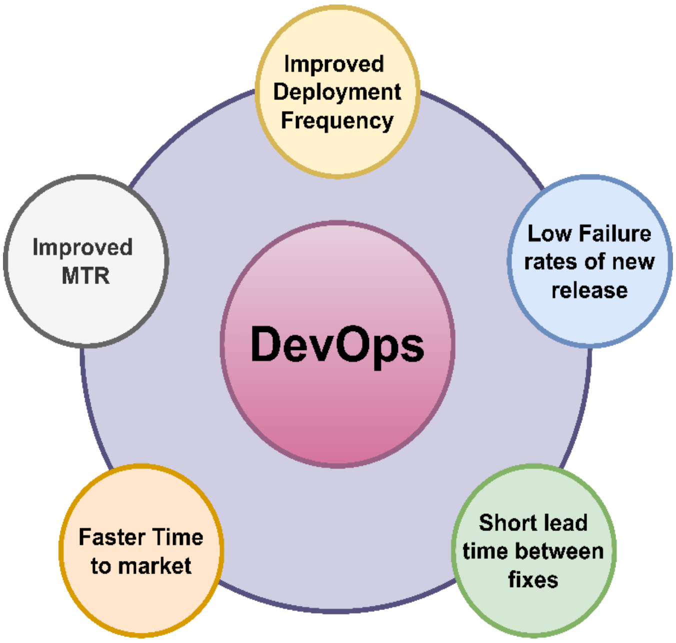
***Key Words*: DevOps, Software Engineering, Agile Methodology, Collaborating, Automation**

1. INTRODUCTION

It's hard to distinguish between DevOps and Agile approaches. This is also due to the fact that many marketers employ broad definitions for these terms, reducing their actual meaning. Furthermore, it is not always marketing ambiguity that causes idea ignorance and confusion. It also contributes to the misconception that DevOps and Agile are more concerned with ideals than with practical actions. A software project requires the completion of activities in order to create additional value. Project management's principal task would be in the planning, design, implementation, and monitoring of these tasks. In terms of software and product design, the business environment has grown more quickly in recent years. Project planning, review, improvement management, and quality control are just a few of the issues that separate successful from unsuccessful initiatives. To address these difficulties, agile approaches are widely adopted and applied around the world. Agile and DevOps techniques are improving methodologies, and as a result, their application development is handled in short space of time (sprints) and other involving team members are changed. The purpose of this research will be on a systematic study of DevOps and agile technique approaches.

2. AN AGILE METHODOLOGY

Software developers are clear of the drawbacks of massive processes like the Waterfall technique, and they are looking for a technique to simplify and make software modification more flexible. They've decided to provide end-user feedback more options to guarantee they're on the proper pathway. In the 1990s, hundreds of new lightweight ideas for software development were evolved, including the well-known Scrum and Kanban approaches. The Agile Software Development Protocol, issued in 2001, defined many of these ideas and placed them within the context of agile software development. The agile technique also challenges the concept of a "completed product," which was the Waterfall approach's primary objective. Agile also enables adaptive planning, continuous improvement, early and continuous delivery, and continuous improvement to help developers to adjust quickly and flexibly to changes in customer demand, software, and other external factors. Agile underlines the value of iterative and incremental software development. This implies that with each new software update, the customer can try out new features or enhance existing ones. Software design can be broken down into smaller chunks called "user stories" using agile methodology.


Improved Deployment Frequency
Low Failure rates of new release
Short lead time between fixes
Faster Time to market
Improved MTR
DevOps

3. THE MODERN DEVOPS

Agile was a solution to waterfall methods' approaches, whereas DevOps was not a response to Agile. Both concepts are not the same, because companies have started to notice synergies and increased performance and efficiency when they are employed together. IT operations (ITOps) and Development Operations (DevOps) are two significant domains that are necessary for organizations in the modern time. DevOps is in service of designing and delivering new products to end clients, although ITOP is in charge of making sure safety, compliance, and availability. DevOps goes hand in hand with adaptability, thorough analysis, and interaction with the implementation of new software, much as ITOps delivers stability and safety for all client needs via the network. Multiple connections between these two aspects emerged as a result of the organization's developments. The focus of Agile Teams is on automation design, automated testing, and Continuous Delivery. These tools, as well as metrics and monitoring systems, configuration management, virtualization, and cloud computing, are regularly used by DevOps teams, according to data. For software engineers who had been dissatisfied by the restrictions of a waterfall technique, Agile appeared to be a completely new concept. Agile isn't perfect. Missed deadlines, completed pieces that are incompatible with each other due to segregated sprints or groups, and additional additions dividing old features – due to a lack of collaboration with DevOps and ITOps – are all common drawbacks of Agile planning.

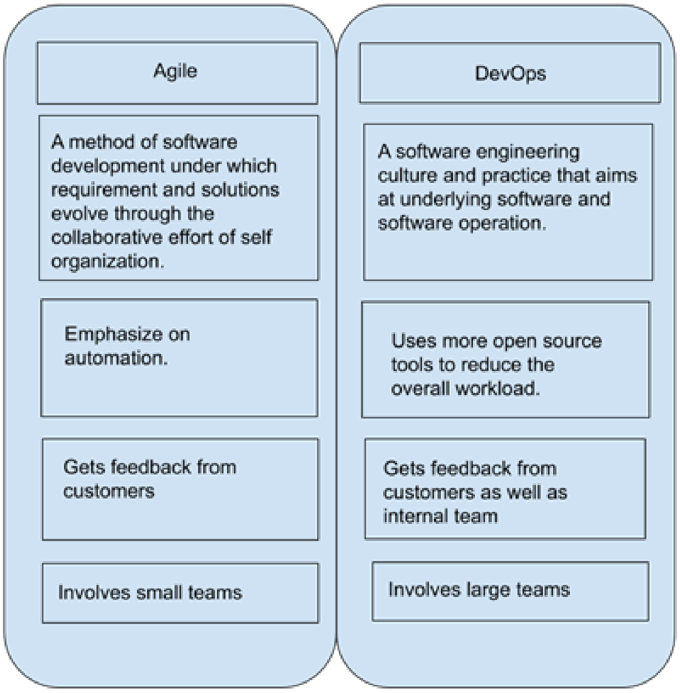
In this circumstance, DevOps fills in the gaps. Because developers must work together, DevOps is a communication strategy that works both within and beyond departments. DevOps communicates with ITOps on a daily basis to keep testing environments safe and efficient, so it makes sense to take the help of other departments, such as marketing and customer support, when deploying new applications. DevOps can be called an enhancement of Agile methodology when both theories are applied to effective business requirements. Agile focuses on cross-functional teams, which are typically made up of Designers, testers, and programmers are all needed

4. DEVOPS VS AGILE

After understanding the backdrop and conversation dimensions, it's time to look at the parallels between DevOps and agile approaches. DevOps is a combination of cultures, techniques, and instruments that splits silos into teams. Agile methodology is a software development approach that accelerates the response loop between clients and software engineers. Although DevOps and agile approaches had such similarities, they are not identical, and some argue that DevOps is preferable to agile methodology. To clear up the confusion, it's crucial to really go down to a nuts and bolts. DevOps establishes the basic organization behavior and environment from a technology standpoint. In this light, the balanced relationship is recognized. Because of the contrasts in both development and production circumstances, well-informed stake holders and phases will be meaningless if the underlying facilities collapse. Global corporate structures, concurrent design, sector - wide management, and adaptive development, for example, can affect future technological innovation.

The difference between the two is defined by what happens following software development. Software is developed, tested, and deployed in both DevOps and Agile. However, after these three stages, pure agility appears to come to an end. DevOps, on the other hand, involves continuous operations. As a result, software development and monitoring are ongoing. Individuals build, test, and deliver software in an agile environment. Technology is business, and software is business, and DevOps is all important for the function of engineers in DevOps. Agile is more associated with lean and resource efficiency, while DevOps is more associated with cost savings, and phrases like agile accounts for projects and a market - ready product (MVP) are key. DevOps brings IT to the mix, but Agile focusses on software development and deployment. Both are significant approaches for software development.

Table -1: Sample Table format



Agile	DevOps
A method of software development under which requirement and solutions evolve through the collaborative effort of self organization.	A software engineering culture and practice that aims at underlying software and software operation.
Emphasize on automation.	Uses more open source tools to reduce the overall workload.
Gets feedback from customers	Gets feedback from customers as well as internal team
Involves small teams	Involves large teams

5. CHALLENGES AND APPROACH

Agile and devops cultural revolutions shift organizational culture from traditional approaches to agile and devops practices. Our research has identified a list of forty distinct barriers between the traditional culture and the agile-devops culture. These are the lack of an agile vision, complexity, legacy systems and legacy software development process, design of large software system, a time to market pressure, planning and risk management, performance, software system cost estimation, quality assurance and measurement in product development, individual roles within the team, strict organization structure with high number of hierarchies and multiple departments for different types of expertise. A list of opportunities to overcome these barriers was compiled and presented as best practices that would guide organizations through their cultural transition from traditional body shop operations to enterprise agile-devops operations.

If you are planning to apply agile and DevOps, you can feel free to consider thehow:It is important to remember that the agile approach is not everything, especially in the case of a large company. It may be difficult to apply Agile: Consider what the challenges are of applying agile mentality to certain parts of your business. Ask yourself how many people will be involved in transforming into an agile company or adopting a DevOps culture? Will it have an impact on your customers? Also ask whether this process can help with innovation. Some companies also request feedback from their customers about what they think about implementing these changes. Overall, if you follow our guide and look at many aspects of your business, you may discover that some benefits will appear in different areas.

Agile software development has been around since 2001, but most organizations do not still adopt this methodology because they are unable to overcome the challenges involved. When a DevOps approach is combined with agile's manifesto and its principles, it leads to continuous improvement and service delivery. The agile method uses quick incremental growth to quickly satisfy customers, thereby increasing product value. This method is based on interactions, customer collaboration, and responding to change. On the other hand, DevOps culture helps in solving problems that can build solutions for the customer's experience as well as increase IT efficiency.

Sample paragraph Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper.

6. CONCLUSIONS

Agile practices and DevOps culture are different, albeit related. Both help teams work faster and more efficiently, where the quality of work is delivered and the customers are satisfied. When effectively practiced, work-life balance and productivity are enhanced [20]. Agile uses incremental and iterative methods to deliver high-quality software in a timely manner. DevOps is a collaboration-building framework for the Technology and Operations team that allows consumers to quickly, automatically deploy code and add code to production. It helps speed up the delivery of products and services across the organization. With these insights in mind, it is clear that both Agile and DevOps strive for a more efficient way to bring value to end-users— but from various angles. It can be defined as aligning development As the use of agile techniques has become pervasive in software development, it has also helped to promote innovative thinking beyond agile development and cross-functional (DevOps) as well as end-to-end processes that impact both software companies and consumers. Several organizations have effectively adopted agile techniques to accelerate the development of applications. The days of waterfall strategies are long gone. Agile has started to slowly expand into other downstream areas, such as 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 entire 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.

REFERENCES

[1] Dingsøyr, 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.

[23] Rico, D., Sayani, H., Sone, S., & Safari, an O'Reilly Media Company, 2009, "The Business Value of Agile Software Methods," J. Ross Publishing.

[24] Conboy, K., Coyle, S., Wang, X., and Pikkarainen, M., 2011, "People over Process: Key Challenges in Agile Development", IEEE Software, 28(4), pp. 48-57.