# Combining Agile with DevOps


Parth Patel

Department of Computer Science and Engineering,
Devang Patel Institute of Advance Technology and Research (DEPSTAR),
Charotar University of Science and Technology (CHARUSAT) ,
Anand, India.
parth.642001@gmail.com

Poojan Vadaliya

Department of Computer Science and Engineering,
Devang Patel Institute of Advance Technology and Research (DEPSTAR),
Charotar University of Science and Technology (CHARUSAT) ,
Anand, India
poojanvadaliya@gmail.com

Nidhi Acharya

Department of Computer Engineering,
Devang Patel Institute of Advance Technology and Research (DEPSTAR),
Faculty of Technology and Engineering (FTE),
Charotar University of Science and Technology (CHARUSAT),
Anand, India.
nidhi.acharya15@gmail.com

Brijeshkumar .Y. Panchal

Department of Computer Science and Engineering,
Devang Patel Institute of Advance Technology and Research (DEPSTAR),
Faculty of Technology and Engineering (FTE),
Charotar University of Science and Technology (CHARUSAT) ,
Anand, India.
panchalbrijesh02@gmail.com

Dweepna Garg

Department of Computer Engineering,
Devang Patel Institute of Advance Technology and Research (DEPSTAR),
Faculty of Technology and Engineering (FTE),
Charotar University of Science and Technology (CHARUSAT) ,
Anand, India.
dweepnagarg.ce@charusat.ac.in

Amit Ganatra

Department of Computer Engineering,
Devang Patel Institute of Advance Technology and Research (DEPSTAR),
Faculty of Technology and Engineering (FTE), Charotar University of Science and Technology (CHARUSAT) ,
Anand, India.
amitganatra.ce@charusat.ac.in



***Abstract*—Despite the fact that agile software development has become progressively pervasive, numerous corporations, institutions, and conglomerates have discovered deficiencies in it. They are unable to offer frequent releases in large part attributable to the involvement of several departments in the process. As a result, many entities have transitioned to DevOps because it is more conducive to it. Given its growing prominence, DevOps is also associated with agile and continuous software development delivery approaches. Because the major goal of agile development is speed, and DevOps is about accuracy, it's no surprise that combining the two would result in the best of both worlds, with teams swiftly building, testing, and deploying updates that are stable and marked improvements over previous builds. We have recognized foundational disorganization in the cooperation and collaboration among the team involved in the projects, which can be solved by DevOps and agile combined. We believe that cooperating with various people who have varied behavioural and functional qualities may lead to better delivery of the project compared to agile.**



***Keywords— Agile, Dev ops, Process, Development, Combination.***


## I. BACKGROUND

Before understanding their combination, we need to first understand the basic terms used, they are agile and devops. Initially, both devops and agile were standalone concepts. Agile is a time-boxed, iterative approach to software development that constructs software incrementally from the beginning of a project rather than attempting to deliver it all at once near the end. It works by breaking down tasks into little slices of user functionality referred as user stories, ranking them, and then delivering them in short sprints known as iterations. While, DevOps is a combination of the words "development" and "operations." It's a set of strategies and technologies aimed to help a company provide applications and services more quickly than traditional software development procedures. Organizations can effectively satisfy their clients and compete in the market because of this quickness. DevOps, in perhaps the most basic form, is about narrowing the

gap between traditionally segregated teams, development, and operations. Development and operations teams collaborate throughout the software application life cycle, from development and testing to deployment and operations, in a DevOps paradigm. Ambler defined that DevOps is an effective collaboration between development teams and operations teams by incorporating the processes [1]. DeGrandiss suggested that DevOps means applying agile processes to operational activities [2].

## II. AGILE AND DEVOPS

Agile is a catch-all word for a wide range of development methodologies based on agile philosophy. Agile attempts to improve software development efficiency and flexibility while avoiding superfluous paperwork and useless work. Agile was first applied in small-scale businesses, but it has recently been adopted by large-scale organisations and projects, including distributed software development. The adoption of agile methodologies articles reflects this shift as well. Agile approaches have been proved in past research to boost customer satisfaction and allow flexible answers to potential difficulties during the development process. It's also been shown to increase project success rates.

We went with Jabbari's definition. DevOps is a development methodology aimed at bridging the gap between Development and Operations, emphasizing communication and collaboration, continuous integration, quality assurance and delivery with automated deployment utilizing a set of development practices. [3] In general, the term DevOps is a combination of development teams and operations teams (or Developers and Operators) [3], and its purpose is bridging the gap between development teams and operations teams [4] DevOps implementation also considers software scalability, how effectively it can be deployed, and how well it can be monitored and maintained in the future. Traditional DevOps benefits, on the other hand, have a drawback. The system lacks the kind of ongoing testing and improvement that Agile provides.

## III. DIFFERNECE BETWEEN AGILE AND DEVOPS

TABLE.1

| **Agile** | **Dev ops** |
|---|---|
| Agile focusses on delivering small results faster. Software is developed in the iterations where team releases new version in short cycles with non-incremental updates | devops usually describe software development and delivery practices practices based on cooperation between developers and operation specialists. |
| Agile targets end-users and developers | devops targets developers and operations team |
| that agile is outward oriented | devops is inward oriented |
| Agile applies to software developers and project managers | devops applies to intersection of development, Quality Assurance and operations |
| Agile has lots of management frameworks to achieve its flexibility and transparency | while no such particular frameworks are applicable for devops |
| The main source of feedback is the end customer | feedback from stakeholders and team is at higher priority |
| Agile mainly focusses on software development | devops focusses more on deployment and post-release alongside development |
| Agile prioritizes flexibility in task | devops prioritizes project documentation |
| Agile projects are difficult to evaluate as requirements are changing | while devops risk arrive from misunderstanding of the term and the lack of suitable tools |

Thus, agile and devops have slightly different focuses and scopes, but the key values are almost identical.

## IV. DRAWBACKS OF AGILE

Agile is currently one of the most popular development methods. It was adopted by many companies, but Agile mainly focused on software development part, and the part of deployment, operation and maintenance always responsible by product operations team or outsourcing company [5]. Using Agile without addressing deployment and operations can cause serious problems [5] and even hinder new features from being presented to users as quickly as possible [6]. In this case, many companies consider adopting DevOps in Agile to attempt to improve the situation [5, 6, 7, 8].

### *A. Inadequate resource planning*

It's hard to forecast efforts like cost, time, and resources required at the beginning of a project,

and this challenge becomes more pronounced as projects get bigger and more complex. Agile is based on the idea that teams can't be sure what their final outcome and perhaps even a few cycles of delivery down the line will look like from day one.

*B. Limited documentation*

Documentation occurs throughout an Agile project, and is routinely "just in time" for generating the output, rather than from the start. As a result, it becomes less detailed and often drops to the bottom of the priority list.

*C. Output that is fragmented*

Incremental delivery may help get goods to market faster, but it's also a major drawback of Agile. This is because when teams work on distinct components in different cycles, the final product is often fragmented rather than a single entity.

*D. There is no conclusion in sight*

It's easy to get sidetracked offering new, unanticipated functionality because Agile needs less planning at the start. Furthermore, because there is never a clear idea of what the "finished result" looks like, projects have no end.

*E. Difficult measurement*

Because Agile works in chunks, tracking progress demands monitoring over multiple cycles. You won't be able to set many KPIs at the outset because the project is "see-as-you-go." It's impossible to keep track of your progress due to the game's duration.

## V. ADOPTING AGILE WITH DEV OPS

More and more people are concerned about Agile and DevOps, and many companies in their project development process also tend to use Agile [15]. But with the continuous development of the software industry, in a short time to the product to market, while maintaining quality, customer expectations and the use of new technology have become the mainstream [8]. Actually, Agile enables fast delivery of software, but it cannot deployment and operational as soon as possible [6].

The agile team mainly has three roles: product owner, master and development team [16]. In the common situation, the development team members need responsible for the development and test. And in order to make their new features or modify part sends to the customer as soon as possible, they need a special operations team [5]. In this case, for solving these questions, many companies consider adopting DevOps to try to improve the situation [6, 7, 8]. DevOps is a development approach designed to bridge the gap between development and operations [18]. One benefit of DevOps is the accelerated feedback loop [13]. With timely feedback from the endusers, software developers deliver products that may be more satisfying and even surprising [14].

Agile and DevOps are becoming increasingly popular, and many firms employ Agile in their project development processes. However, with the continued evolution of the software industry, bringing a product to market in a short amount of time while maintaining quality, meeting customer expectations, and utilising new technologies have become the norm. Agile allows for rapid software delivery, but it cannot be deployed and operational as quickly as possible. Product owner, master, and development team are the three major positions in an agile team. In most cases, members of the development team are responsible for both development and testing. And they'll need a specific operations team to get their new features or modified parts to the consumer as soon as possible. Despite the fact that DevOps holds great promise for closing the gap between development and operations, incorporating this paradigm into Agile may be difficult. When agile methods are projected to be applied, for example, a significant amount of redesign for the maintenance work process is frequently required, and all studies show that this is difficult to achieve.

## VI. FEW CONSIDERATIONS IN ORDER TO SUPPORT DEV OPS WITH AGILE

Although DevOps is very promising to eliminate the gap between development and operations, passing this method is not easy. At present, some companies using DevOps in Agile have encountered some problems [9, 10, 11, 12, 13].

- There should be sustained collaboration between the operations and development teams throughout the solution development lifecycle. This aids are fully comprehending the business strategy, release schedules, and assessing the technical and scheduling feasibility of the solution in order to confirm the Ops-related needs. Release, service, and change management, environment provisioning,

application deployment, and automation and tools should all be covered by DevOps teams

- The DevOps team must assist the Product Owner in comprehending non-functional requirements (NFRs) so that the development team can incorporate them into the final product architecture and solution. Technical issues such as deployment and support platforms, vendor dependencies, and third-party interfaces/applications that are required to produce the final solution should be considered and conveyed to the relevant stakeholders.

- When using agile to execute DevOps, it's also important to consider the product backlog for NFRs, the technical requirements for deployment and maintenance, performance requirements, rollback and roll forward rules, and, most importantly, security and firewall requirements.

- To ensure that any dependencies, feature alignments, and product upgrades are realistically accounted for, the DevOps team should be included in Sprint backlog planning and daily stand-ups, Sprint review, Scrum, and plan alignment. The development team's ongoing interaction and communication with the operations team also keeps the ops team informed about which functionality release timelines are approaching. The operations team may then assist the development team in more accurately planning the release schedule and assisting the development team in shipping the product faster.

- Greater automation is needed to reduce the requirement for coordination between infrastructure and development teams.

## VII. BENEFITS OF COMBINING AGILE WITH DEV OPS

Combining DevOps with Agile Integration has a lot of advantages. Some of them are:

- Processes for releasing products are streamlined, and higher-quality products are delivered.
- It makes it easier to collaborate.
- In each release, there is more value and fewer risks.
- There are fewer bugs and they are fixed faster.
- Visibility has improved.
- Greater customer loyalty.

## VIII. FACTORS TO PONDER UPON WHILE COMBINING DEV OPS AND AGILE

Some of the most typical stumbling blocks encountered when integrating DevOps into Agile development are listed here, along with solutions.

### A. Coherent Teamwork

- For team members, the DevOps architecture and Agile approach will provide a deeper understanding of all development variables. It facilitates and transparent communication.

- Software functionality, delivery, and maintenance should be considered by all team members involved in the development process. The service, management, environment provisioning, release cycles, automation tools, and application integration will all be better understood by the teams.

- Agile offers practicality to the team, while DevOps adds business value.

### B. Interpreting Software Lifecycle

- When DevOps principles are introduced at the start of the development cycle, everyone on the team saves time and resources. There will be fewer changes and fewer errors as a result. DevOps and Agile work together to ensure consistency and accelerate the time to market for a product or service

### C. DevOps Adoption in Sprints

- Agile workflow believes that the software development process is separated into sprints, hence integrating DevOps management into sprint management is critical.

- As you begin to incorporate the DevOps methodology into your Sprints, follow these recommendations.

- Invite members of the operations, infrastructure, and support teams to help plan sessions.
- Discuss the product's functioning and usability.
- They should be included in the upcoming sprint.
- Participate in sprint backlog preparation, daily meetings, sprint review, and scrum and plan alignment with the DevOps team.
- Your development team's participation and collaboration with your operations team also keeps the Ops team informed about functionality release dates. The Ops team may then support the development team in more accurately planning the release schedule and in shipping the product faster.

D. Quality Assurance

- When merging DevOps and Agile, QA/Quality assurance is a must. At each level, frequent testing will rule out any likelihood of errors. This will improve the software's performance and load testing. Smaller release cycles and shorter time to market result from continuous development.

E. Service Backlog

- When integrating DevOps and Agile, service backlogging is a prerequisite. The following elements must be present in a DevOps structure. Integration and Scalability of Software are included in it. Monitoring, logging, and alerting capabilities are all available as part of the service. Details on security and compliance is also included.

F. Automation

- When combining Agile and DevOps, automating workflow is highly encouraged. Any potential flaws will be eliminated by automating code scanning processes. To make release cycles easier, artefacts should be saved in a repository. The teams' total productivity will improve, and there will be fewer opportunities for errors.

G. Documentation

- DevOps teams document the entire process until the product is released, whereas Agile teams do not record minutes of meetings or other interactions. It's a good idea to keep a record of everything for future reference.

H. Measurement and Analysis

- To understand the performance and progress of the workflow after using Agile and DevOps, you must keep track of key KPIs. This will also make it easier for you to set up many successful release cycles. At the same time, it is time and cost effective. The following are the few parameters that we need to keep track of, according to the Scrum Alliance Organization. From start to finish, the time it takes to complete a project. The percentage of people who keep their release dates. Increase in the number of people who have been released by a certain percentage. Requirements for assistance and defects can be found on any platform.

## IX. POSITIVE CASE STUDY ON CERTAIN COMPANIES

### A. AMAZON

When Amazon operates on dedicated servers, it has always been a challenge to predict the amount of equipment that needs to be purchased to meet traffic demand and generate quotes to handle unexpected traffic spikes. .. As a result, about 40% of Amazon's server capacity was wasted, and when traffic tripled during the holiday shopping season, more than three-quarters were unused and wasted money on purchases. ..

After online retailers moved to the Amazon Web Services (AWS) cloud, engineers were able to gradually increase or decrease capacity. Not only did this reduce server capacity spending, but it also facilitated the transition to a continuous deployment process, allowing developers to deploy their own code to the servers they need, whenever they need it.

Within a year of migrating from Amazon to AWS, engineers deployed code every 11.7 seconds on average. Agile approaches have also reduced both the number and duration of outages, resulting in increased revenue.

B. NETFLIX

Netflix was breaking new ground when Netflix evolved its business model from DVD shipments to streaming video over the Internet. It turned into an opensource solution because there were no commercial tools available to keep the company's vast cloud infrastructure running smoothly. Simian Army was created with the help of hundreds of developer volunteers. It's a set of automated tools that stress-test your Netflix infrastructure and allow you to proactively identify and fix vulnerabilities before they impact your customers.

Since then, Netflix has been working on automation and open source, and today engineers deploy code thousands of times a day. This year, Netflix was unanimously selected for the JAX Special Jury Award, and JAXenter editor Coman Hamilton said, "The speed at which this entertainment game changer adopts new technology and implements it in the DevOps approach is new to IT. We are setting standards. "

C. TARGET

Within Target, several groups have been working on DevOps for years. According to technical architect Dan Cundiff, "what started in a small corner of the development and infrastructure teams has since spread like a wildfire."

He is not exaggerating. Today, DevOps not only promoted the development of projects like Target's mobile savings app, Cartwheel, but also changed the culture. Target is currently hosting DevOps Days for in-house teams with demos, open labs, lightning talks, breakout sessions, and guest keynotes. The company continues to spread good news to the business community by sponsoring the DevOps Days conference in Minneapolis.

D. WALMART

Walmart is the king of major retailers in the heart of the United States, but has always struggled online behind the Amazon. To lay the foundation, the company formed a state-of-the-art team through multiple technology acquisitions and in 2011 founded Walmart Labs, a retailer's innovation and development division.

Walmart Labs takes a dedicated DevOps approach to its mission. It included cloud-based One Ops technology that automates and accelerates application delivery. It also includes Hapi, a Node.js framework for building applications and services that allows developers to focus on creating reusable application logic rather than spending time building infrastructure. We have developed several opensource tools. Recently, the company has continued to evolve its agile approach by deploying over 100,000 OpenStack cores to build its own private cloud.

E. NORDSTORM

Nordstrom's growth model still included waterfall delivery, large batch releases, and many shared services as they began rewriting their in-store customer management application. When he finally launched the program two and a half years later in 2011, it was no longer relevant. “It was a big wake-up call for us as an organization that we needed to find a way to stay in the environment.

Nordstrom's customer mobile app team pioneered the DevOps redesign. After explaining mobile device 22-28 week delivery times, the team bridged the gap between product development and support and the teams organized around value. The company also switched to continuous planning and moved to single backlog. As a result, errors decrease, throughput increases, and releases go from twice a year to monthly. More importantly, Nordstrom realized that these methods could work for any team and continued to apply them across the organization.

F. FACEBOOK (NOW META)

Facebook contributed to a shift in how we think about software development. In all but name, many of the principles it adopted early on, such as code ownership, incremental modifications, automation, and continuous improvement, were DevOps. It has refined its strategy over time, and it just moved its whole infrastructure and back-end IT to the Chef configuration management platform (and made some of its cookbooks available to the public).

Consumer expectations of software are being reshaped by Facebook's fast development lifecycle. Its recent announcement of biweekly app upgrades successfully issued notice that regular, rapid app refreshes are the new normal, and any company that can't keep up risks being left behind.

## X. CONCLUSION

The world is rapidly changing so are the requirements of the clients. The agile methodology aims at providing the solution to these ever-changing requirements. But it lacks in certain aspects, mainly on the part of deployment and maintenance. Dev ops comes to the rescue for this problem. So, in this case study we reflected about how the combination of agile and dev ops can reduces the demerits of agile and also how it will and

has impacted the industry. We also highlighted certain aspects to consider and some attention needed for the combination of these two methodologies.

## XI. REFERENCES


[1] Scott W Ambler. Disciplined agile delivery and collaborative devops. Cutter IT Journal, 24(12):18, 2011.

[2] Dominica DeGrandis. Devops: So you say you want a revolution? Cutter IT Journal, 24(8):34, 2011.

[3] Ramtin Jabbari, Nauman bin Ali, Kai Petersen, and Binish Tanveer. What is devops?: A systematic mapping study on definitions and practices. In Proceedings of the Scientific Workshop Proceedings of XP2016, XP '16 Workshops, pages 12:1–12:11, New York, NY, USA, 2016. ACM.

[4] Johannes Wettinger, Uwe Breitenbücher, and Frank Leymann. Devopslang – bridging the gap between development and operations. In Massimo Villari, Wolf Zimmermann, and Kung-Kiu Lau, editors, Service-Oriented and Cloud Computing, pages 108–122, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[5] J. Sutherland and R. Frohman. Hitting the wall: What to do when high performing scrum teams overwhelm operations and infrastructure. In 2011 44th Hawaii International Conference on System Sciences, pages 1–6.

[6] Airaj Mohammed. Enable cloud devops approach for industry and higher education. Concurrency and Computation: Practice and Experience, 29(5):e3937. e3937 CPE-15-0382.R1.

[7] Marta Olszewska and Marina Waldén. Devops meets formal modelling in highcriticality complex systems. In Proceedings of the 1st International Workshop on Quality-Aware DevOps, QUDOS 2015, pages 7–12, New York, NY, USA, 2015. ACM.

[8] Soon K. Bang, Sam Chung, Young Choh, and Marc Dupuis. 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, RIIT '13, pages 61–62, New York, NY, USA, 2013. ACM

[9] Erich F. M. A., Amrit C., and Daneva M. A qualitative study of devops usage in practice. Journal of Software: Evolution and Process, 29(6):e1885. e1885 smr.1885.

[10] E. Diel, S. Marczak, and D. S. Cruzes. Communication challenges and strategies in distributed devops. In 2016 IEEE 11th International Conference on Global Software Engineering (ICGSE), pages 24–28, Aug 2016.

[11] L. E. Lwakatare, T. Karvonen, T. Sauvola, P. Kuvaja, H. H. Olsson, J. Bosch, and M. Oivo. Towards devops in the embedded systems domain: Why is it so hard? In 2016 49th Hawaii International Conference on System Sciences (HICSS), pages 5437–5446, Jan 2016.

[12] Jens Smeds, Kristian Nybom, and Ivan Porres. Devops: A definition and perceived adoption impediments. In Casper Lassenius, Torgeir Dingsøyr, and Maria Paasivaara, editors, Agile Processes in Software Engineering and Extreme Programming, pages 166–177, Cham, 2015. Springer International Publishing.

[13] Leah Riungu-Kalliosaari, Simo Mäkinen, Lucy Ellen Lwakatare, Juha Tiihonen, and Tomi Männistö. Devops adoption benefits and challenges in practice: A case study. In Pekka Abrahamsson, Andreas Jedlitschka, Anh Nguyen Duc, Michael Felderer, Sousuke Amasaki, and Tommi Mikkonen, editors, ProductFocused Software Process Improvement, pages 590–597, Cham, 2016. Springer International Publishing.

[14] Dana Pylayeva. Introduction to DevOps with Chocolate, LEGO and Scrum Game. Apress

[15] S. Sharma and N. Hasteer. A comprehensive study on state of scrum development. In 2016 International Conference on Computing, Communication and Automation (ICCCA), pages 867–872, April 2016.

[16] Howard Lei, Farnaz Ganjeizadeh, Pradeep Kumar Jayachandran, and Pinar Ozcan. A statistical analysis of the effects of scrum and kanban on software development projects. Robotics and Computer-Integrated Manufacturing, 43:59 – 67, 2017. Special Issue: Extended Papers Selected from FAIM 2014.