DevOps vs Agile

Axel Lárusson

April 2019

1 Introduction

In the ever changing world of technology new ideas are introduced daily and it can be a difficult task to be on top of all the buzzwords, new tools, new approaches or work-flows. Some have turned into industry standards, like: Waterfall, Agile (scrum, kanban, hybrids), Rapid Application Development (RAD), Six Sigma, and many, many more. Now recently there has been a new methodology that is gaining significant traction and that is "DevOps". Some people have claimed that this new methodology will be the next industry standard and completely take over from most of the older methods.

> *"Organizations that have adopted DevOps are more likely to deploy on demand and prioritize automation than those practicing Agile."[1]*

Then there are others who view this new methodology (DevOps) as normal evolution of previous methods and believe that they need to work together.

> *"DevOps is Agile for the Rest of the Company"[2]*

> *"DevOps is a natural extension of Agile, some say DevOps 'began' when teams started extending Agile principles to the infrastructure, and more specifically, to systems administration."[3]*

Depending on where you look, there are many people, teams and companies who have their own idea of what DevOps is exactly and how best to adopt this methodology. In order to better differentiate between Agile and DevOps it is necessary to look into their fundamental characteristics.

2 What is Agile?

I think it is good to start at the beginning and go all the way back to 2001 when the Agile methodology was introduced in the Agile manifesto. In this manifesto the Agile methodology was described as four values and twelve principles.[4]

Values:

1. Individuals and Interactions Over Processes and Tools
2. Working Software Over Comprehensive Documentation
3. Customer Collaboration Over Contract Negotiation
4. Responding to Change Over Following a Plan

Principles:

1. Customer satisfaction through early and continuous software delivery
2. Accommodate changing requirements throughout the development process
3. Frequent delivery of working software
4. Collaboration between the business stakeholders and developers throughout the project
5. Support, trust, and motivate the people involved
6. Enable face-to-face interactions
7. Working software is the primary measure of progress
8. Agile processes to support a consistent development pace
9. Attention to technical detail and design enhances agility
10. Simplicity
11. Self-organizing teams encourage great architectures, requirements, and designs
12. Regular reflections on how to become more effective

Agile is in essence a software development approach that follows the values and principles of the Agile methodology. The focus is on team oriented, iterative development and good communication both within the team and with the customer. It is important to be flexible and adaptable to new tools and trends on the market and have the ability to change tools and approaches even late in the development process.

While Agile focuses mainly on the space between the developer and the customer there was not much focus on the space between the developer and operations. This space has also been growing in more recent times with the rise of the microservice architecture. When the industry standard architecture changes, from monolithic architecture to microservice architecture there is a rising demand to synchronize teams within the organization and provide an overview for the end-to-end engineering process. This is where DevOps comes in.

3 What is DevOps?

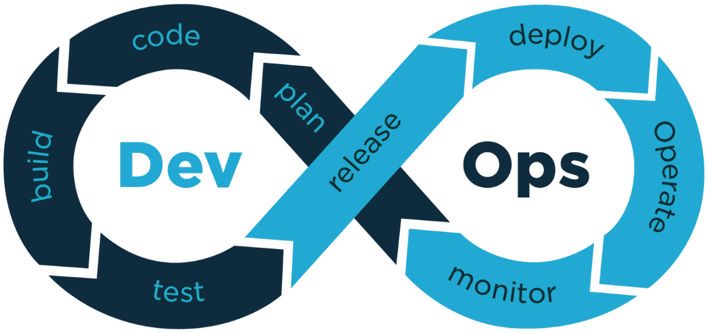
Similar to Agile, DevOps is a software development methodology but the focus is different. DevOps aims to bring software development teams(Dev) and information technology operatives(Ops) together. The idea of DevOps aims to adapt a culture where a company actively breaks down silos and moves towards a streamlined software development process. Also, DevOps aims to change and improve the relationship between these two business units by advocating better communications and collaboration.

The main goals of DevOps are to:

- Improve deployment frequency
- Achieve faster time to market
- Lower failure rate of new release
- Shorten lead time between fixes
- Improve mean time to recovery

3.1 DevOps lifecycle

DevOps can be difficult to understand, especially if people are not technically skilled or have limited knowledge about the subject. Therefore the DevOps lifecycle was introduced to better explain the process visually.[5]



Plan: Here we figure out what application we are developing and how we are going to approach that task. Here we also create a rough picture of the development process. Usually the first step when creating a new application and then used as a milestone in the development process.

Code : Here is where the application is made by developers. The coding follows the plan from earlier step.

Build : Here is where the application is built from the code in the previous step.

Test : Here is where we test the application, see if everything works and rebuild if necessary. This is regarded as the most important step in the cycle.

Release : If an application succeeds in the test phase then it gets released. Here is where an application is checked and packed, ready for deployment.

Deploy : Here is where we actually deploy our application to our users.

Operate : Here is where you can configure and manage your application that has been deployed. Often called *"Configure"*.

Monitor : Here we monitor the performance of our application. We can investigate performance and make modifications if they are needed.

DevOps also allows Agile development teams to implement Continuous Integration, Continuous Delivery and Continuous Deployment. These methods help them to automate and streamline their processes.

Continuous Integration:

This is the process of automating the build and test phases from the DevOps lifecycle. This is considered to be the most important component(handles tests, which is considered the most important phase in the DevOps lifecycle) in an effective DevOps development culture. Continuous Integration puts great emphasis on testing automation and prevents from broken builds and conflicts between teams.

Continuous Delivery:

We use Continuous Delivery when we want to streamline and automate the processes that lead up to the deployment of our application. Here we focus on quality of builds, test performance of builds and find ways to remove bottlenecks and other issues.

Continuous Deployment:

With Continuous Deployment we go further down the DevOps lifecycle. We automate the process that if a application passes all the phases of the production pipeline it is released to the customer without a human intervention. Continuous Deployment is great to accelerate feedback from the customer and it removes the concept of a *"Release Day"*.

Implementing DevOps Tools

Today there are a number of great tools that organizations can integrate to their software development process to take advantages of the DevOps methodology. Example tools: Jenkins, Travis, Docker, CircleCI, GitLab CI and many more.

4 Differences and Similarities

Now that we have established that Agile and DevOps are both a software development methodologies, it might be a good idea to explore further what are the differences and similarities between those two methodologies.

4.1 Differences

Methodology:

Agile development team is only involved in building the software and launching. DevOps focuses on deploying software in the most reliable and safest way, which isn't always the fastest.

Approach:

Agile focuses mainly on speed and good communication with the customer to provide efficient iterative software development, Agile teams work fast, using short sprints and scrum meetings to deal with much time pressure. DevOps focuses mainly on operational and business readiness, with DevOps there is a focus on the long-term schedule with minimum work disruption. DevOps is highly dependent on communication but there are fewer meetings.

Teams:

Agile usually works in small teams that can tackle complex problems and deliver fast results. Agile development emphasizes on training all team members to obtain many similar skills. The developers are trained to do many different types of work and therefore become well rounded programmers. This is great when someone in the team needs help, the team gets a new team member or something goes wrong. Then the team can help each other and increase productivity and let team members learn new things in their work. DevOps is focused on many teams working together as a larger unit with the same goal in mind. With this mindset it is better to specialize on your part of the development process.

Communication:

Agile focuses on variety of formal and informal meetings and feedback is given by the customer. For DevOps the documentation is crucial since feedback comes from the internal team.

Documentation:

Agile teams use meetings to synchronize and don't focus much on documentation. DevOps teams on the other hand need to be able to understand every software, new and old. To tackle this Devops relay on thorough documentation.

4.2 Similarities

Both are methodologies to enhance efficiency

Both Agile and DevOps are methodologies that aim toward greater productivity for an organization. By adapting either one or both into an organization will bring with it improvement in productivity, automation, break down silos within an organization and many more positive effects.

Both want to be a part of the *"Company Culture"*

Both Agile and DevOps work best if they are not regarded as a process within an organization but rather a common understanding between peers as a company culture. Teams follow these processes because it is the culture not because they were told to enforce them by an authoritative figure in the company.

Frequent collaboration

DevOps and Agile share a common goal when it comes to collaboration. For both methodologies there is much emphasis on collaboration between team members.

5 Discussion

When it comes down to choosing which methodology an organization wants to implement in to their software development process it always comes down to preferences. While Agile and Devops have different uses, focus and implementations, they both aim to move an organization more from the centralized model and distribute the responsibilities to teams. This makes the developer more independent and makes the team that follows either Agile, DevOps or both to be more productive.

There have been much debate in the IT industry about this topic and many try to put these two methodologies against each other to find out which is better and which one is better for a team to implement. The Agile approach encourages teams to break down large problems and work from smaller more manageable problems that when put together lead to a large change. Most companies in the IT industry have already implemented Agile and it's considered the software development methodology standard today.

But Agile is not perfect and has many problems when facing outer complications, for example: missed deadlines, incompatible software and new feature breaking old functions. This is where DevOps helps. The DevOps methodology is rooted in communication, both within teams and between departments within an organization. DevOps relies on frequent communications between developers and operations to ensure secure and stable develop environments. Many believe once a company gets to big for an effective agile approach then DevOps will replace Agile. While that idea is valid for some cases the general approach should be to use both. It is better to view DevOps as an extention of Agile. While Agile relies on teams that are cross-functional and have various software developers, DevOps takes that idea further by adding the operations in to the process. By doing so you close the gap between teams and make the process from software development to deployment more manageable. Because DevOps emphases communication it can also improve overview and transparency for all teams.[6]

> *"Both DevOps and Agile can work in tandem since they can complement each other. DevOps promotes a fully automated continuous integration and deployment pipeline to enable frequent releases, while Agile provides the ability to rapidly adapt to the changing requirements and better collaboration between different smaller teams."[7]*

With DevOps, organizations can now shift their deployment cycles to days and weeks instead of years. DevOps also offers maintainability, predictability and monitoring of the development pipeline where the operation team are fully aware of the progress of the development team.

DevOps allows Agile development teams to implement Continuous Integration, Continuous Delivery and Continuous Deployment. This helps them to launch products faster into the market with more automation. Automation helps teams to accelerate software development and deployment processes to get their application to customers faster.

6 Conclusion

What we can conclude from this paper is that Agile is basically a methodology for developing applications, while the DevOps mainly emphasize on application deployment. It can be very difficult to compare these two methodologies because they work on different methods with different approaches.
It has been shown that by actively deciding to implement both an organization can benefit greatly as it leads to more rational decision making.[6] By using both Agile and DevOps an organization can get the best of both worlds.

References

[1] "2018 Global Developer Report." https://about.gitlab.com/developer-survey/2018/ [Accessed: Apr 30 2019], 2018.

[2] T. Bradlay, "Devops is agile for the rest of the company." https://devops.com/devops-is-agile-for-the-rest-of-the-company/ [Accessed: Apr 30 2019], March 2015.

[3] K. Casey, "Agile vs. devops: What's the difference?." https://enterprisersproject.com/article/2019/1/agile-vs-devops-whats-difference [Accessed: Apr 30 2019], January 2019.

[4] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, R. C. Martin, S. Mellor, D. Thoma, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, K. Schwaber, J. Sutherland, and D. Thomas, "Manifesto for agile software development." https://www.agilealliance.org/agile101/the-agile-manifesto/ [Accessed: Apr 30 2019], 2001.

[5] Guru99, "Devops tutorial," 2019.

[6] S. Watts and C. Kidd, "Devops vs agile: What's the difference and how are they related?." https://www.bmc.com/blogs/devops-vs-agile-whats-the-difference-and-how-are-they-related/ [Accessed: Apr 30 2019], August 2017.

[7] K. Ismail, "Agile vs devops: What's the difference?." https://www.cmswire.com/information-management/Agile-vs-devops-whats-the-difference/ [Accessed: Apr 30 2019], November 2018.