

The History of DevOps

From the article I read, it seems that DevOps was created anytime between 2007 and 2008. DevOps was up and coming and no one really cared for it. The big “popping” thing to do was agile methodology. With the new methodology everything was great until things needed to be done together. There was so much dysfunction that was happening with keeping different departments separate but only come together at the end when things are “finished”. That led to unhappy customers, which then led to the blame game, which then led to long hours of fixing the problem. No one likes redoing work that was already “done” so I know it led to unhappy people and turned the organizations upside down. So, with the DevOps coming in people started to combine departments and stopped having different buildings. This led to people working together and in turn made it easier to find the mistakes that were made before and fix them before releasing the product/software to customers. DevOps helped organizations to stop waiting on upper management for changes and let the teams make small changes and automate them to see when, where, and who made the changes. This helped cut down on long hours and made customers happier as well.

DevOps is still a work in progress, not every team has mastered it. Some organizations still have trouble with it, but all that matters is trying. If the teams try to always have continuous feedback, iterations and improvement to deploy better and faster customer needs then they are on their way to a better future.

The Lean Movement

Lean software development is an agile framework that is based on eliminating waste, optimization of development time and resources, and delivering on what the product needs. The lean approach is also known as the Minimal Viable Product (MVP) strategy. The whole approach to this lean method is to give the customer a basic release of the product and let the customer give feedback on what they want to add and want to remove. Then after the feedback is taken the team then works on the product to give them what they want.

The whole reason for this movement is to stop overusing resources for things that do not get used or eventually get tossed. This also makes it so where the whole team works together and also gets the work done faster.

The Agile Manifesto

So, the Agile Manifesto was created in 2001 and is a part of the Agile Methodology. There are 12 principles that the Manifesto lives by. Those 12 principles are:

1. satisfy the customer through the early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity—the art of maximizing the amount of work not done—is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

The Continuous Delivery Movement

With the Continuous Delivery method, it allows for developers to already automate the code, and have it tested before it is even implemented into the code. This allows developers to already know if the implementation is going to be beneficial or not. It also allows developers to not have to wait on the upper management to approve the test then see if it works. Developers can now show the work and automate it to the system, which creates a faster and better work environment but also better response time for customers. This way of working delivers faster updates, finds bugs and addresses them faster and improves developer productivity.

Resources

<https://www.atlassian.com/devops/what-is-devops/history-of-devops>

<https://www.productplan.com/glossary/lean-software-development/>

<https://aws.amazon.com/devops/continuous-delivery/>

<https://agilemanifesto.org/principles.html>

<https://www.atlassian.com/agile/manifesto>