

The History of DevOps

The Beginning

“Despite the rise of agile methodology, development and operations teams remained siloed for years. DevOps is the next evolution of collaboration tools and practices to release better software, Faster” (Buchanan). Between 2007 and 2008 a new technological movement was beginning to gain momentum, right when IT operations and software development communities raised concerns about what they felt was a fatal level of dysfunction in the industry. Through the discussions between Andrew Clay and Patrick Debois, the concept of DevOps emerged, it was their concerns about the drawbacks of Agile that lead them to create a new Development Methodology. In 2009 an event was held in Belgium, “DevOpsDays”, where it gained quite a lot of popularity. ***“DevOps implementation ensures that developers can now take part in deployment, admins can write scripts and QA engineers know how to figure out issues other than testing. Processes can be automated and no one has to wait as they can now work more closely and come up with quicker and better solutions”*** (Agarwal).

The Lean Movement

Lean methodology is a concept that began decades before the digital era, and continues to benefit all tech industries. Lean methodology began around 1913 tracing back to Henry Ford and Kiichiro Toyoda. Henry Ford was an innovator, with one of his notable innovations being the “Flow Production” process. The Flow Production allowed Ford to assemble Model Ts rapidly, which is exactly what the Lean methodology is all about ***“a way of optimizing the people, resources, effort, and energy of your organization toward creating value for the customer. It is based on two guiding tenets, continuous improvement and respect for people”*** (Lynn).

The Agile Manifesto

The Agile Manifesto is a document that helps to identify four of the key values, and 12 principles that its authors believe software developers should use to guide their work. The manifesto was created by 17 developers in 2001. Calling themselves the Agile Alliance, they sought an alternative to existing software development processes because they saw them as complicated, unresponsive processes that were too focused on documentation requirements. The Agile

Alliance wanted **“to restore credibility to the word methodology”** (Pratt). The Agile Manifesto contains four core values of Agile, being:

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

With these four values, the Manifesto also contains 12 principles:

1. Satisfying customers through early and continuous delivery of valuable work
2. Breaking big work down into smaller tasks that can be completed quickly
3. Recognizing the best work emerges from self-organized teams
4. Providing motivated individuals with the environment and support they need and trusting them to get the job done.
5. Creating processes that promote sustainable efforts
6. Maintaining a constant pace for completed work
7. Welcoming changing requirements, even late in a project
8. Assembling the project team and business owners on a daily basis throughout the project
9. Having the team reflect at regular intervals on how to become more effective, then tuning and adjusting behavior accordingly
10. Measuring progress by the amount of complete work
11. Continually seeking excellence
12. Harnessing change for a competitive advantage

The four values outlined in the Manifesto promote a software development process that focuses on quality products that meet consumers expectations, while the 12 principles are intended to create and support a work environment that is focused on the customer, and aligns with business objectives.

The Continuous Delivery Movement

“Continuous delivery is a software development practice where code changes are automatically prepared for a release to production. A pillar of modern application development, continuous delivery expands upon continuous integration by deploying all code changes to a testing environment and/or a production environment after the build stage. When properly implemented, developers will always have a deployment ready build artifact that has passed through a standardized test process” (Freeman). With continuous delivery developers can automate testing, allowing them to go beyond just unit tests. With cloud computing it is easy and cost-effective to automate the creation and replication of multiple environments for testing.

Agarwal, H. (2022, June 28). *Roadmap to it revolution: History of devops: Appknox*. Secure Mobile App & Data with Mobile App Security Testing. Retrieved August 8, 2022, from <https://www.appknox.com/blog/history-of-devops>

Buchanan, I. (n.d.). *History of DevOps*. Atlassian. Retrieved August 8, 2022, from <https://www.atlassian.com/devops/what-is-devops/history-of-devops>

Freeman, E. (2019). *DevOps*. Amazon. Retrieved August 8, 2022, from <https://aws.amazon.com/devops/continuous-delivery/>

Insights, D. (2021, December 9). *Lean, Agile, and DevOps: A focus on delivering value*. 3Pillar Global. Retrieved August 8, 2022, from <https://www.3pillarglobal.com/insights/lean-agile-and-devops-a-focus-on-delivering-value/>

Lynn, R. (2021, May 28). *Lean methodology*. Planview. Retrieved August 8, 2022, from <https://www.planview.com/resources/articles/lean-methodology/#:~:text=In%20short%2C%20Lean%20methodology%20is,improvement%20and%20respect%20for%20people>

Pratt, M. K., & Torode, C. (2020, April 7). *What is the Agile Manifesto?* SearchCIO. Retrieved August 8, 2022, from <https://www.techtarget.com/searchcio/definition/Agile-Manifesto>