The History of DevOps

Matthew R. Howard

Bellevue University

Table of Contents

[The History of DevOps 3](#_Toc519178863)

[The Lean Movement 3](#_Toc519178864)

[The Agile Manifesto 3](#_Toc519178865)

[The Continuous Delivery Movement 4](#_Toc519178866)

[References 5](#_Toc519178867)

The History of DevOps

DevOps began in 2008 out of a need to streamline and quickly develop applications. By using DevOps, it’s been shown that “High-performing IT organizations experience 60x fewer failures and recover from failure 168x faster than lower-performing IT organizations. Additionally, they can deploy 30x more frequently with 200x shorter development schedule.” (Searle, 2016)

# The Lean Movement

The idea of the Lean Startup is to keep you inventory at a minimum. This means having them minimum amount of supplies you need on hand to do the job. It also means to keep the order que down to a minimum as well. Orders should almost immediately go into fulfillment once the order is received. This will maximize efficiency in the manufacturing (developing) process. This has been a best practice of Toyota for years before it came to the world of developers (Mezak, 2018). In IT, The Lean Startup begins your product development by giving a clear idea of how you should develop your product and what features should change and how (Marschall, 2012). Additionally, after having your ideas down, metrics of how well they’re being implemented need to be captured (Marschall, 2012).

# The Agile Manifesto

The Agile Manifesto came in the 1990s out of the frustrations of huge lag time between business requirements and the delivery of the technology that would answer the need for said business requirements (i.e. the applications and feature the customer was requesting). This was due in part to the Waterfall Model, which was the standard for developing software at the time. The Waterfall Model wasn’t meeting the demand for more products in a shorter time and didn’t take advantage of how quickly software can be changed. Because of this lag time, requirements, customer circumstances, and businesses changed which led to many of those projects to be cancelled because the final product didn’t meet the need of the request. Then in 2000, seventeen developers came together to devise a better, faster way to develop software called Agile (SmartSheet, 2018). The Agile Manifesto has four principal values:

1. Individual 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

Using these four foundation principals Agile can align development with business needs while encouraging customer participation and feedback allowing for a better product to be made in faster time and to the customers’ needs (SmartSheet, 2018).

The Continuous Delivery Movement

Continuous Delivery focuses on transforming the technical aspects of software delivery and is critical in achieving successful DevOps practices. Continuous Delivery is a subset of Agile, however, it does not involve stopping to make special effort to create a releasable build. Nor is it about adopting a shorter cycle for making software ready for release. It’s keeping the software ready for release at all times during development (Morris, 2012). Continuous Delivery doesn’t require frequent releases, instead, it only requires ensuring software could be released with little effort at any point during the development. There are, however, issues that come up between Continuous Delivery and Agile and that is: the code should not have any incomplete stories or bugfixes at the end of the iteration (Morris, 2012). This comes from the idea that at the end of the iteration, the team does extra work to make prepare the product for release. But in Continuous Delivery, no additional work is needed to make the software releasable. Using feature toggles allows the team to meet the requirement of the build and be ready for release at the end of the iteration, even with unfinished stories (Morris, 2012). The bottom line is that Continuous Delivery offers developers the following:

1. Decreased risk by uncovering deployment issues earlier.
2. Increased flexibility by giving the organization the option to release at any time with minimial added cost or risk.
3. Involves everyone at all times in the production release.
4. Improves the quality of the software by forcing teams to fix problems as they’re found.

References

Mezak, S. (2018, January 24). The Origins of DevOps: What's in a Name? Retrieved July 12, 2018, from <https://devops.com/the-origins-of-devops-whats-in-a-name/>

Searle, W. (2016, November 09). What is DevOps and Where Did it come from? A Brief History. Retrieved July 12, 2018, from <https://turbonomic.com/blog/on-turbonomic/a-brief-history-of-devops/>

Marschall, M. (2012, July 19). Devops and The Lean Startup. Retrieved July 12, 2018, from <https://www.agileweboperations.com/devops-and-the-lean-startup>

SmartSheet. (2018, February 12). Comprehensive Guide to the Agile Manifesto. Retrieved July 12, 2018, from <https://www.smartsheet.com/comprehensive-guide-values-principles-agile-manifesto>

Morris, K. (2012, May 09). Continuous Delivery vs. Traditional Agile - DZone DevOps. Retrieved July 12, 2018, from <https://dzone.com/articles/continuous-delivery-vs>