Eric Williams-Phillips

Bellevue University | CSD 370

Sue Sampson

**History of DevOps**

DevOps is an approach to making the software development process as efficient as possible. Before DevOps, Waterfall was the go-to method for planning, organizing, and developing software. While it isn’t necessarily a bad process, it was noticeably slow when used for larger projects and with larger teams. There was little flexibility, and it was too difficult to make changes to software during the process. This brought about the need for a new methodology to optimize the development cycle. Agile was an approach that emerged in the 90s and improved on some of the weaknesses of Waterfall. Using Agile, developers worked in sectioned “sprints” that allowed for more changes to be made during the cycle, collaboration with customers, and reduced risk. This alone, however, did not fully solve some of the major issues organizations were running into when it came to developing and deploying software. The development and operations departments were typically kept separate and did not always synergize well with one another. DevOps was coined by Patrick Debois in 2009 to merge the two practices for a more reliable software delivery plan.

**The Lean Movement**

Lean software development is all about reducing wasteful usage of resources during the development process. It is based on the Lean manufacturing principles of the 80s created by James Womack and Daniel Jones. The principles revolve around the idea of identifying and mapping valuable customer needs, creating a flow of value in the manufacturing process, transitioning from pushing work on people to allowing people to pull work, and ultimately reducing the time it takes to delivery value to customers. The Lean software development process upholds this idea of reducing wastefulness by avoiding unnecessary code, limiting WIP, and experimentation before commitment. Lean development practices have helped with collaboration in that teams and individuals share knowledge and work together in decision-making. It also incorporates a constant line of communication with the customer to enable continuous improvement.

**The Agile Manifesto**

The Agile Manifesto is a document created in 2001 by a group of 17 people that were unhappy with the quality of software being produced by many companies. They believed these companies were losing sight of the true goal of customer satisfaction. The document that was produced by this group, The Agile Manifesto, listed 12 principles that would go on to “change software development forever” (Atlassian, n.d.). The principles in the document aim to value:

* Individuals and interactions
* Working software over documentation
* Customer collaboration
* Response to change

According to the document itself, customer satisfaction takes the highest priority (*Principles behind the Agile Manifesto*, n.d.). Over 20 years later, experts still see this manifesto as relevant. An Agile Coach for LinkedIn, Tanner Wortham, says that he still quotes the manifesto and even used its principles during his time in the Marines. The CEO of Scrum.org said that “agile principles aren’t really new at all” and that “They are just being applied in a different way” (Atlassian, n.d.).

**The Continuous Delivery Movement**

According to IBM, continuous delivery aligns with the first principle of the Agile Manifesto—customer satisfaction —by focusing on efficiently moving code into production. This involves automation at most steps of the development lifecycle. Automation frees up time and provides continuous testing, saves costs, scales into bigger projects, and helps deploy code into each phase of the software development process. In fact, IBM recommends the continuous delivery process be as automated as possible to create a “good, reliable delivery pipeline” (*What Is Continuous Delivery? | IBM*, n.d.). The Agile methodology uses continuous delivery by releasing features to the customer as they become available. This approach also satisfies the Agile Manifesto by providing continuous delivery and “harnessing change for the customer’s competitive advantage” (*Principles behind the Agile Manifesto*, n.d.).

Resources:

Atlassian. (n.d.-a). *History of DevOps | Atlassian*. https://www.atlassian.com/devops/what-is-devops/history-of-devops

Atlassian. (n.d.). *Is the Agile Manifesto Still a Thing? | Atlassian*. <https://www.atlassian.com/agile/manifesto>

Lutkevich, B., & Silverthorne, V. (2021, December 6). *Lean software development*. Software Quality. https://www.techtarget.com/searchsoftwarequality/definition/lean-programming

*Principles behind the Agile Manifesto*. (n.d.). <https://agilemanifesto.org/principles.html>

Walker, J. (2022, October 26). Development vs Operations: Where Do &quot;Dev&quot; and &quot;Ops&quot; Roles Differ and Overlap? *How-To Geek*. https://www.howtogeek.com/devops/development-vs-operations-where-do-dev-and-ops-roles-differ-and-overlap/

*What is Continuous Delivery? | IBM*. (n.d.). https://www.ibm.com/topics/continuous-delivery