Rachel Theis

The History of DevOps

CSD 380

6.1.25

**Introduction**

DevOps refers to the practice of applying the most tried and true principles of manufacturing and other physical sectors to IT organizations and projects (Kim et al., 2016). DevOps does not refer to just one methodology, but rather a combination of many trusted and successful processes and systems, including Lean, Agile, and the Continuous Delivery Movement (Kim et al., 2016). The foundational methodologies of DevOps are Lean, the Theory of Constraints, and the Toyota Kata Movement (Kim et al., 2016). However, DevOps doesn’t only consider preexisting non-technical methods, it also expands upon earlier software-originated processes, namely Agile (Kim et al., 2016).

**The Lean Movement**

The Lean Movement is the critical precursing methodology to DevOps. The first modern example of Lean processes is seen in 1913 with Henry Ford’s automobile production, where flow production was established by using interchangeable parts and a moving conveyor belt (Lean Enterprise Institute, 2023). However, Ford’s Lean processes had a major weakness – the inability to provide variety and options to customers (Lean Enterprise Institute, 2023). Starting in the 1930s, the Toyota automobile company started looking for ways to implement Ford’s Lean practices while meeting customer needs (Lean Enterprise Institute, 2023). Toyota created the Toyota Production System, which shifted to focus on the flow of the system as a whole rather than on individual machines (Lean Enterprise Institute, 2023).

Lean techniques like Value Stream Mapping, Kanban Boards, and Total Productive Maintenance were initially used in the manufacturing sector, including being codified for Toyota’s systems in the 1980s (Kim et al., 2016). In 1990, authors Womack, Roos, and Jones established the five principles of Lean in their book, *The Machine that Changed the World*. In 1997, the Lean Enterprise Institute began researching how Lean principles could be applied at organizations outside the manufacturing sector (Kim et al., 2016).

Five Principles of Lean (Do, 2017):

1. Define Value: Value is what the customer is willing to pay for. It is crucial to understand what customers want and what they are willing to pay for, even in the customer doesn’t have a clear picture of this, which is common with new or complicated technology (Do, 2017).
2. Map the Value Stream: Using the defined value, the next step is to map all activities that contribute to value. Non-value-adding activates are considered waste, which should be eliminated when possible or reduced as much as possible (Do, 2017).
3. Create Flow: This step is to ensure that processes run smoothly and breakdowns or delays are mitigated. Flowing strategies include reconfiguring production steps, leveling out workload, creating cross-functional departments, and training employees to be multi-skilled (Do, 2017).
4. Establish Flow: The goal of this step is to minimize the number of work-in-progress (WIP) projects, while ensuring the materials and resources are available for work. This seeks to establish a pull system with just-in-time deliveries (Do, 2017).
5. Pursue Perfection: This step includes establishing a culture of continuous learning and improvement. The organization should seek to always be improving, and employees should be invested in delivering perfection and meeting customer needs (Do, 2017).

**The Agile Manifesto**

The Agile Manifesto was published in 2001 by 17 of the leading voices in software development (Kim et al., 2016). The Agile Manifesto aims to provide a lightweight set of principles and guidance that can better serve projects than heavyweight processes and methods like Waterfall and the Rational Unified Process (Kim et al., 2016). DevOps and Agile don’t occur in a vacuum, and several modern DevOps discoveries or key moments have occurred at Agile conferences or within the Agile community (Kim et al., 2016).

Agile’s inception wasn’t necessarily presenting new or unused ideas to software development. Instead, they codified these ideas that were informally being used across teams and organizations in software development, and had been used in other sectors before the software boom (Drumond, n.d.). One contributor of the Agile manifesto stated, “When I look at the tenets behind the manifesto, these aren’t tenets that were invented by us,” West said. “They’re the tenets of the scientific method. Galileo used them. Archimedes used them.” (Drumond, n.d.)

“…perhaps what's important isn't one blessed document that everyone can agree on, but whether or not a group of people (from a team to an entire organization) can apply the ideas in the Manifesto to their specific situation without losing sight of its spirit. And if we can do that well, the possibilities are unlimited.” (Drumond, n.d.).

**The Continuous Delivery Movement**

The Continuous Delivery Movement originated in 2006 and defines the role of a “development pipeline” which calls for constantly deployable code and infrastructure (Kim et al., 2016). The Continuous Delivery Movement utilizes an iterative feedback loop that emphasizes delivering to the customer as quickly as possible, learning from hands-on experience, and incorporating feedback into the next release (Kim et al., 2016). Benefits of the Continuous Delivery Movement include automating the software release process, improving developer productivity, finding and addressing bugs sooner, and delivering updates faster (AWS, n.d.). The Continuous Delivery Movement uses central DevOps, Agile, and Lean principles like flowing processes to deliver results to customers as quickly as possible and valuing and implementing user feedback (Atlassian, n.d.).

Key Continuous Delivery Principles (Atlassian, n.d.):

1. Business Value of Continuous Delivery: Improves productivity, velocity, and sustainability of projects
2. Value Steam Mapping: This technique can be used to enhance the value of continuous delivery.
3. Repeatable, Reliable Processes: Reliable consistency between environments and executions eliminates certain avoidable bugs
4. Automate Everything: Automation is a core principle of continuous delivery. The team should automate as much as possible, including tests, releases, and configuration changes.
5. Version Control: Version control is an essential tool for successful continuous delivery.
6. Built-in Quality: The iterative feedback loop keeps quality in mind and as a priority throughout the project’s lifecycle.
7. Do the hardest part first: Pain points in a project should be addressed first to mitigate weaknesses and avoid these challenges from growing worse over time. Additionally, painful yet repeatable processes may be able to be solved through automation, especially if done first.
8. Everyone is responsible: The entire team owns the continuous delivery and can support these processes throughout their roles.

**References**

Atlassian. (n.d.). *Continuous Delivery Principles*. Atlassian. <https://www.atlassian.com/continuous-delivery/principles>

AWS. (n.d.). *What is Continuous Delivery? – Amazon Web Services*. Amazon Web Services, Inc. <https://aws.amazon.com/devops/continuous-delivery/>

Do, D. (2017, August 5). *The Five Principles of Lean*. The Lean Way. <https://theleanway.net/The-Five-Principles-of-Lean>

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

Kim, G., Humble, J., DeBois, P., Willis, J. (2016). *The DevOps Handbook*. IT Revolution Press, LLC.

Lean Enterprise Institute. (2023). *A Brief History of Lean*. Lean Enterprise Institute. <https://www.lean.org/explore-lean/a-brief-history-of-lean/>