Alisa Steensen 5/29/25

Module 1.3 DevOps

DevOps movement began between 2007 and 2008. Software developers felt there was dysfunction within the system and needed something to combat this. This went against the traditional software development model. Developers and IT professionals often times had competing objectives, separate department leadership, and separate performance indicators in which they were judged. Because of this teams were isolated. After online forums and local meetups began to pop up regarding the subject, DevOps was created. DevOps created an environment of continuous feedback and collaboration between all.

DevOps is a part of each phase in the development and operations lifecycle. Beginning with the planning stage to the building, monitoring, and iterating phases. DevOps brings together the skills, processes, and tools from every facet of an engineering and IT organization. (Atlassian n.d.) By utilizing Agile methodologies, teams can better plan and produce work being done. Continuous integration and delivery are a cornerstone of DevOps practices which removes the likelihood of bugs or incidents. Additionally, DevOps enables the automation of version control capabilities, which are fundamental. While utilizing IT teams to manage the end-to-end delivery of IT services to customers, it better delivers consistent software. Overall, these processes lead to improved collaboration and faster deployment times, which ultimately lead to higher quality products.

The Lean Movement

The Lean Movement originated from Toyota’s production system and became the Lean Software Development. This process emphasizes minimizing waste (such as unnecessary code, features, and processes). Additionally, it encourages continuous improvement, fast feedback loops, and delivering only what provides value to the user. Instead of following rigid procedures, Lean Software Develpment promotes focusing on what the customer truly needs. Developers are urged to avoid over complicating designs and simplify the workflow. The main goal is to create high quality software by streamlining the process while always keeping the user in mind. These principles lay a foundation for more flexible and efficient workflows, which shift teams away from waterfall style development process. Lean Software development is the main reason for why DevOps and Agile works efficiently today.

The Agile Manifesto

The Agile Manifesto originated in 2001 and was crated by 17 software practitioners who wanted a more flexible and collaborative approach to development processes. It emphazises individuals and interactions with working software through customer collaboration. The Agile methodologies have became the core principles of the Manifesto and remain pertinent. The Agile Manifesto came about to combat the traditional and rigid development practices such as Waterfall model. The Manifesto contains twelve core values:

* Satisfy the customer through continuous delivery.
* Welcome changing requirements, even late in development.
* Providing and delivering working software frequently.
* Business people and developers must work together daily.
* Build projects around motivated individuals, trust them to get the job done.
* Have face to face conversations for an efficient and effective method.
* Working software is the primary measure of progress.
* Sponsors, developers, and users should be able to maintain a constant pace.
* Continuous attention to technical excellence and good design.
* Keeping it simple.
* Self organizing teams provide the best architectures, requirements and designs.
* Teams reflect on how to become more effective and changes behavior accordingly.

The Agile Manifesto remains a highly influential piece of software development today and beyond. Frameworks such as Scrum, Kanban, and SAFe are built on Agile principles. And the manifesto isn’t a rulebook, it is a mindset to follow. Organizations that follow the Agile principles build faster with improved products, with better collaboration across departments.

The Continuous Delivery Movement

The Continuous Delivery Movement is a way to get new code ready to live automatically. Continuous delivery emphasizes automation that minimizes the manual effort required to deploy code changes. In a continuous delivery pipeline, developers create their code, it passes automated tests, and is then automatically integrated with a repository like a container registry or a binary repository. (CodeFresh n.d.) This helps developers prepare updates so they can be sent to users without a lot of manual steps. When it’s used with continuous integration, this process is called CI/CD. With CI in place, developers can safely move their code into testing and even into the final product because it has already passed the important checks. This helps catch problems early and keep things running smoothly.

Continuous Delivery additionally ensures that there are automatic tests done on code, not small unit tests, but also checks things like buttons and features, seeing how fast a system handles alot of traffic, testing how parts of the app work together, and more. Because of this process, it makes it easier to catch bugs before users see them. CI/CD helps developers ship better products with simpler releases, easier maintenance, better development, better quality, and less downtime. This ultimately leads to better customer satisfaction.

In summary, the DevOps movement developed as a response to the communication gap and issues happening between software developers and IT operations. It has brought about a technology shift in how software is built, tested, and delivered. Influenced by the Lean Movement, Agile Manifesto, and Continuous Delivery, DevOps promotes fast feedback and collaboration across all stages of the software development lifecycle. By embracing these practices, organizations can better deliver products faster and easier, while adapting to change.

References:

Atlassian. (n.d.). *History of DevOps*. Retrieved May 29, 2025, from <https://www.atlassian.com/devops/what-is-devops/history-of-devops>

Soares, L. (2021, August 13). *Towards lean software development*. Medium. <https://medium.com/codex/towards-lean-software-development-24460340b11a>

Atlassian. (n.d.). *Agile Manifesto for Software Development*. Retrieved May 29, 2025, from <https://www.atlassian.com/agile/manifesto>

Agile Alliance. (n.d.). *12 principles behind the Agile Manifesto*. Retrieved May 29, 2025, from <https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/>

Codefresh. (n.d.). *What is continuous delivery and how does it work?* Retrieved May 29, 2025, from <https://codefresh.io/learn/continuous-delivery/>