

Nima Memarzadeh

Nathan Braun

CSD380

05/28/2025

## The History of DevOps

Software development evolved through its continuous pursuit of enhanced value delivery while advancing quality and increasing change response speed. DevOps represents the convergence of software development and IT operations and developed gradually over time. It is the product of three influential movements: Three foundational movements including Lean manufacturing alongside Agile development and the Continuous Delivery Movement contributed to DevOps. A comprehensive knowledge of DevOps origins allows one to fully understand its fundamental principles and operational methods.

### **The Lean Movement**

The origins of the Lean movement trace back to early 20th-century manufacturing innovations by Henry Ford and Kiichiro Toyoda. With the introduction of flow production Ford optimized the assembly process to mass-produce vehicles including the Model T through complete integration. Because this model could make only one product type it lacked the ability to produce different versions. Toyoda developed Ford's production approach further by creating the Toyota Production System during the 1950s which included quick setup techniques and self-monitoring systems along with inventory control mechanisms. The Toyota Production System emphasized kaizen or continuous improvement which established the foundation for Lean thinking (3Pillar Global, 2021).

The emphasis on efficiency and quality combined with continuous incremental improvements made Lean a good choice for the ever-changing software development field. Traditional business practices failed to keep up with the rapid advancements of personal computers during the 1990s. The requirement to reduce software delivery timeframes without sacrificing quality led to the emergence of Agile development methods.

## **The Agile Manifesto**

The origin of Agile stemmed from a need to overcome the strictness of conventional software development methods like the waterfall model which required exhaustive documentation and initial planning. The method became obsolete because it couldn't keep pace with the rapidly changing demands of customers. The 1990s saw the emergence of multiple alternative software development methodologies such as Rapid Application Development (RAD), Dynamic Systems Development Method (DSDM), Scrum, and Extreme Programming (XP) which all focused on speed and iterative processes with adaptable frameworks (Sharma, 2023b).

In February 2001 seventeen developers came together to consolidate these concepts into what became known as the Agile Manifesto. The group established four main values along with twelve principles that prioritized working software and collaboration while allowing for flexibility over strict processes (Sharma, 2023b). The Agile framework doesn't remove risk entirely but manages it through swift development iterations and ongoing verification. Through timeboxing, test-driven development, pair programming and retrospectives teams maintained quality while responding to changing priorities.

The evolution of Agile turned it into a movement centered on delivering value through incremental and iterative processes. Agile methodology had to undergo evolution to better connect development activities with operational functions.

## **The Continuous Delivery Movement**

Development teams increased their code output after adopting Agile practices yet deployment processes continued to slow down progress. Operations teams received code without proper communication or shared ownership responsibilities. This disconnect created friction, errors, and delays. Continuous Delivery (CD) alongside Continuous Deployment (CD) represent two approaches designed to automate and simplify the release process.

Continuous Delivery maintains deployment readiness through testing emphasis while also incorporating packaging and staging practices. Continuous integration and small frequent updates work together to minimize risk in this process. In most cases Continuous Delivery requires a manual approval process before deploying to production. Continuous Deployment bypasses the manual release gate by automatically pushing code to production after successful testing according to Sharma (2023c).

Both deployment practices aim to achieve routine and low-risk software releases. These practices require extensive test coverage and transparency to allow teams to track changes as they happen in real time. Embracing the mindset that code remains unfinished until deployment triggered the growth of the DevOps movement which unites development and operations teams across the entire software lifecycle (Sharma, 2023c).

### **The Rise of DevOps**

The late 2000s demonstrated that efficient software delivery demanded closer teamwork between development teams and operations staff. The cultural and professional movement of DevOps evolved from the combination of Agile development practices and operational excellence standards. Gartner states that DevOps seeks to eliminate service delivery challenges by integrating Lean and Agile principles throughout the software development phases (3Pillar Global, 2021).

While Agile centers on development tasks DevOps manages the complete delivery pipeline which includes coding and production system monitoring. The approach centers around automated processes and collective accountability while emphasizing ongoing feedback mechanisms and experimental practices. Through faster and safer software releases DevOps teams achieve reliable delivery which fulfills their ultimate objective of providing user value.

DevOps history represents an amalgamation of multiple complementary movements rather than a one-directional evolution. The Lean approach instilled in us an appreciation for efficient processes alongside ongoing enhancement. Agile emphasized adaptability, collaboration, and speed. Continuous Delivery applied these principles to create an automated release pipeline that operates smoothly. DevOps emerges from these movements as a unified method which enhances communication, automation of processes and delivers customer-centric value. The historical background of DevOps reveals why it represents more than tools and practices as it fundamentally changes the approach to software development and delivery.

## Works Cited

- 3Pillar Global. (2021, July 9). *Lean, Agile, and DevOps: A Focus on Delivering Value*. Retrieved from <https://www.3pillarglobal.com/insights/blog/lean-agile-and-devops-a-focus-on-delivering-value/>
- Sharma, A. (2023b, August 2). *A Brief History of DevOps Part II: Agile Development*. CircleCI. Retrieved from <https://circleci.com/blog/a-brief-history-of-devops-part-ii-agile-development/>
- Sharma, A. (2023c, November 9). *A Brief History of DevOps Part IV: Continuous Delivery vs. Continuous Deployment*. CircleCI. Retrieved from <https://circleci.com/blog/a-brief-history-of-devops-part-iv-continuous-delivery-and-continuous-deployment/>