Kypton Lantz  
August 15, 2025  
CSD380-0307 DevOps (2261-DD)  
Module 1.3 Assignment: The History of DevOps

**The History of DevOps**

DevOps did not appear out of thin air or a toolchain. It emerged around 2009 as a practical response to a chronic problem: software teams could build fast, but getting reliable changes into production was slow and siloed. The term “DevOps” came about around the first DevOpsDays in Ghent, Belgium, organized by Patrick Debois, after momentum from the famous Flickr “10+ deploys per day” talk at Velocity 2009 (Allspaw & Hammond, 2009; Devopsdays, 2009; Devopsdays, n.d.). The point was not tooling; it was making delivery flow safely and at speed. Three earlier movements set the stage: Lean, Agile, and Continuous Delivery.

**The Lean Movement**

Lean started in manufacturing (Toyota Production System) with one central obsession: improve flow of value by relentlessly removing waste and building quality in (jidoka), using small batches, standard work, and continuous improvement (kaizen) (Lean Enterprise Institute, n.d.-a; Lean Enterprise Institute, n.d.-b). These ideas map almost one-for-one to modern delivery:

* **Small batches:** smaller, safer deploys.
* **Limit WIP and smooth flow:** fewer handoffs and less queueing between dev and ops.
* **Visualize work and expose problems quickly:** monitoring, observability, automated tests, and fast rollback.

DevOps is Lean applied to software delivery. Producing a shortened lead time, decreased variability, discovering defects early, and a continually improved system. If you do not reduce batch size and queues or build fast feedback loops, you are not doing DevOps; you are just automating a mess.

**The Agile Manifesto**

Agile (2001) explicitly demands early and continuous delivery of valuable software while responding to change (Agile Alliance, n.d.; Agile Manifesto, n.d.). That vision hits a wall if operations remain a separate, ticket-driven bottleneck. Agile teams wanting frequent, reliable releases needed to expand their scope into build, environments, testing, and operations concerns. In other words, Agile’s values forced the conversation: if you want to deliver frequently, you must make delivery *deliverable*. DevOps is the organizational and technical alignment that lets Agile’s promise show up in production.

**Continuous Delivery Movement**

Jez Humble and David Farley’s Continuous Delivery (CD) codified the engineering practices to keep software always in a deployable state with automated build/test, environment parallelism, deployment automation, and versioning everything (Humble, 2010; Humble, 2010b; Fowler, 2013). CD is not “deploy all the time” (that is continuous deployment); it is “you **can** deploy at any time,” safely, by making release a business decision rather than an IT fire drill (Humble, 2010b). CD operationalizes Lean (small batches, fast feedback) and fulfills Agile (frequent, reliable delivery). Without CD practices, “DevOps” devolves into culture posters and tool sprawl.

**The Turning Point in 2009**

Two events made the implicit explicit:

* **Velocity 2009**: Flickr’s John Allspaw and Paul Hammond showed “10+ deploys per day” driven by dev-ops cooperation, automation, and discipline (Allspaw & Hammond, 2009).
* **DevOpsDays Ghent**: Patrick Debois convened practitioners around breaking the dev/ops wall; the name “DevOps” stuck, and the community expanded globally (Devopsdays, 2009; Devopsdays, n.d.).

From there, community practice, case studies, and research (e.g., DORA/Accelerate) reinforced the core idea: teams that optimize for flow and feedback ship faster and more reliably.

**Conclusion**

* **Lean** gave DevOps the economic and operational logic: flow of value, small batches, fast feedback.
* **Agile** set the delivery goal: frequent, change-friendly releases that actually reach users.
* **Continuous Delivery** provided the concrete mechanics: automate the path to production so every change is releasable.

DevOps is not a replacement for Lean, Agile, or Continuous Delivery, but the convergence of these philosophies and practices. Together, they reshaped how software moves from idea to production.

**References**

Agile Alliance. (n.d.). *Agile manifesto for software development*. <https://agilealliance.org/agile101/the-agile-manifesto/>

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

Allspaw, J., & Hammond, P. (2009). *10+ deploys per day: Dev and Ops cooperation at Flickr* [Conference talk]. O’Reilly Velocity. <https://www.youtube.com/watch?v=LdOe18KhtT4>

Devopsdays. (2009). *Devopsdays Ghent 2009*. <https://legacy.devopsdays.org/events/2009-ghent/>

Devopsdays. (n.d.). *About devopsdays* (History). <https://devopsdays.org/about/>

Fowler, M. (2013, May 30). *Continuous Delivery*. <https://martinfowler.com/bliki/ContinuousDelivery.html>

Humble, J. (2010, August). *Continuous delivery vs continuous deployment*. <https://continuousdelivery.com/2010/08/continuous-delivery-vs-continuous-deployment/>

Lean Enterprise Institute. (n.d.-a). *Toyota Production System (TPS)*. <https://www.lean.org/lexicon-terms/toyota-production-system/>

Lean Enterprise Institute. (n.d.-b). *A brief history of lean*. <https://www.lean.org/explore-lean/a-brief-history-of-lean/>