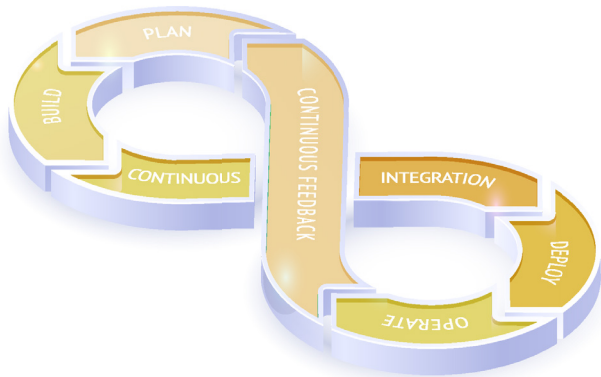


DevOps Ideas, Patterns, and Practices are Central to Agile 2



According to Deutsche Bank CIO Frederic Veron, “enterprises that wish to reap the potentially rich rewards of getting IT and business line leaders to build software together in agile fashion must also embrace the DevOps model.”¹

Why is that? It’s simple: *DevOps is necessary to scale Agile*. DevOps practices are what enable an organization to rapidly deploy changes to many different parts of their product, across many products, on a frequent basis—with confidence.

That last part is key. Companies like Amazon, Google, and Netflix developed DevOps methods so that they could deploy frequently at massive scale without worrying if they will break something. **DevOps is, at its core, a risk management strategy.** DevOps practices are what enable you to maintain a complex multi-product ecosystem and make sure that everything works. DevOps substitutes traditional risk management approaches with what the [Agile 2](#) authors call *real-time risk management*.²

You might think that all this is just for software product companies. But today, most organizations operate on a technology platform, and if you do, then DevOps applies. DevOps methods apply to any enterprise that creates and maintains products and services that are defined by digital artifacts.

DevOps methods apply to any enterprise that creates and maintains products and services that are *defined by digital artifacts*.

That includes manufacturers, online commercial services, government agencies that use custom software to provide services to constituents, and pretty much any large commercial, non-profit, and public sector enterprise today.

As JetBlue and Breeze airlines founder David Neeleman said, “we’re a high-tech company that just happens to fly airplanes,”³ and Capital One Bank’s CIO Rob Alexander said “We’re a founder-led, 20-year-old technology company.”⁴

Most large businesses today are fundamentally technology companies that direct their efforts toward the markets in which they have expertise, assets, and customer relationships.

1 <https://www.cio.com/article/3141577/true-agile-software-development-requires-devops.html>

2 Agile 2: The Next Iteration of Agile, by Cliff Berg et al, pp 205 ff.

3 <https://www.businessinsider.com/breeze-airways-pushing-back-launch-until-2021-what-we-know-2020-7>

4 <https://www.youtube.com/watch?v=0E90-ExySb8>

DevOps Is Necessary at Scale

Scaling frameworks such as SAFe and DA provide potentially useful patterns for organizing the work of lots of teams. However, DevOps is arguably more important than any framework, because without DevOps methods, scaling is not even possible, and many organizations (Google, Amazon, Netflix...) use DevOps methods at scale without a scaling framework.

If teams cannot deploy their changes without stepping on each other's work, they will often be waiting or going no faster than the slowest team, and lots of teams will have a very difficult time managing their dependencies—no framework will remedy that if the technical methods for multi-product dependency management and on-demand deployment at scale are not in place. If you are not using DevOps methods, you cannot scale your use of Agile methods.

How Does Agile 2 View DevOps?

DevOps as it is practiced today is technical. When you automate things so that you can make frequent improvements to your production systems without worrying about a mistake, you are using DevOps. But DevOps is not a specific method. It is a philosophy that emerged over time. In practice, it is a broad set of techniques and approaches that reflect that common philosophy.

With the objective of not worrying in mind, you can derive a whole range of techniques to leverage tools that are available today: cloud services, elastic resources, and approaches that include horizontal scaling, monitoring, high-coverage automated tests, and gradual releases.

While DevOps and Agile seem to overlap, especially philosophically, DevOps techniques are highly technical, while the Agile community has not focused on technical methods for a very long time. Thus, DevOps fills a gap, and Agile 2 promotes the idea that Agile and DevOps go best together.

DevOps evangelist Gene Kim has summarized DevOps by his “Three Ways.”¹ One can paraphrase those as follows:

- 1. Systems thinking:** always consider the whole rather than just the part.
- 2. Use feedback loops** to learn and refine one's artifacts and processes over time.
- 3. Treat everything as an experiment** that you learn from, and adjust accordingly.

The philosophical approaches are very powerful for the DevOps goal of delivering frequent changes with confidence, because (1) a systems view informs you on what might go wrong, (2) feedback loops in the form of tests and automated checks tell you if you hit the mark or are off, and (3) if you view every action as an experiment, then you are ready to adjust so that you then hit the mark. In other words, you have created a self-correcting system.

Agile 2 takes this further by focusing on the entire value creation flow, beginning with strategy and defining the kinds of leadership that are needed. Agile 2 promotes product design and product development as parallel and integrated activities, with feedback from real users and real-world outcomes wherever possible. This approach embeds Gene Kim's three DevOps “ways” into the Agile 2 model, unifying Agile 2 and DevOps.

¹ <https://itrevolution.com/the-three-ways-principles-underpinning-devops/>