

The Evolution of DevOps - The Past, Present and Future (Revised)

Per Arn (perarn@kth.se), Tobias Gabi Goobar (tobiasgg@kth.se)

May 2022

1 Introduction

The term DevOps essentially describes the principles and values one should aim towards following when seeking to develop software and perform IT operations efficiently [3]. Efficiency is a must when developing software and can be achieved in different ways, with DevOps being the common approach for many. DevOps aims to increase efficiency by minimizing the time spent in the life cycle of system development [27] and “provide continuous delivery with high software quality” [26]. Many of the core building blocks of DevOps originate from agile principles such as valuing continuous delivery integration, ensuring to always have a stable build which can be built further upon, stressing the importance of communication and having a neat culture for all stakeholders involved in the projects. A problem with agile software development, which, granted, did thrive after the Agile Revolution, was that although the software development teams were efficient, they lacked communication with the IT operations teams. The solution to this was combining the agile software development with IT operations and thus DevOps originated [3]. This essay aims to show the evolution of DevOps, from its inception in 2008, to the best practices in the industry today and finally some predictions for the future of DevOps. By studying the evolution of DevOps, one can get an understanding of its purpose, through learning about its beginnings and history, and also why the DevOps practices look like they do today. Furthermore, reflecting over the future uncovers deficits in current practices and also gives an idea of what is useful to learn more about to prepare.

2 History of DevOps

2.1 Debois, Shafer and the Agile Systems Administration Group

The concept of DevOps came into existence in 2008 after discussions made by Patrick Debois and Andrew Shafer regarding the drawbacks of Agile at *Agile Infrastructure* conference. They argued that although the agile principles are neat, the drawback is that the development and operations teams are not synchronized enough which introduces unnecessary complexity when developing artifacts. Debois and Shafer formed the *Agile Systems Administration Group*. Although, at this time the word DevOps had yet to be introduced [11] [19].

2.2 10+ Deploys Per Day: Dev and Ops Cooperation at Flickr

At the O'Reilly Velocity conference in 2009 the talk *10+ Deploys Per Day: Dev and Ops Cooperation at Flickr* was given by John Allspaw and Paul Hammond. The talk highlighted the importance of

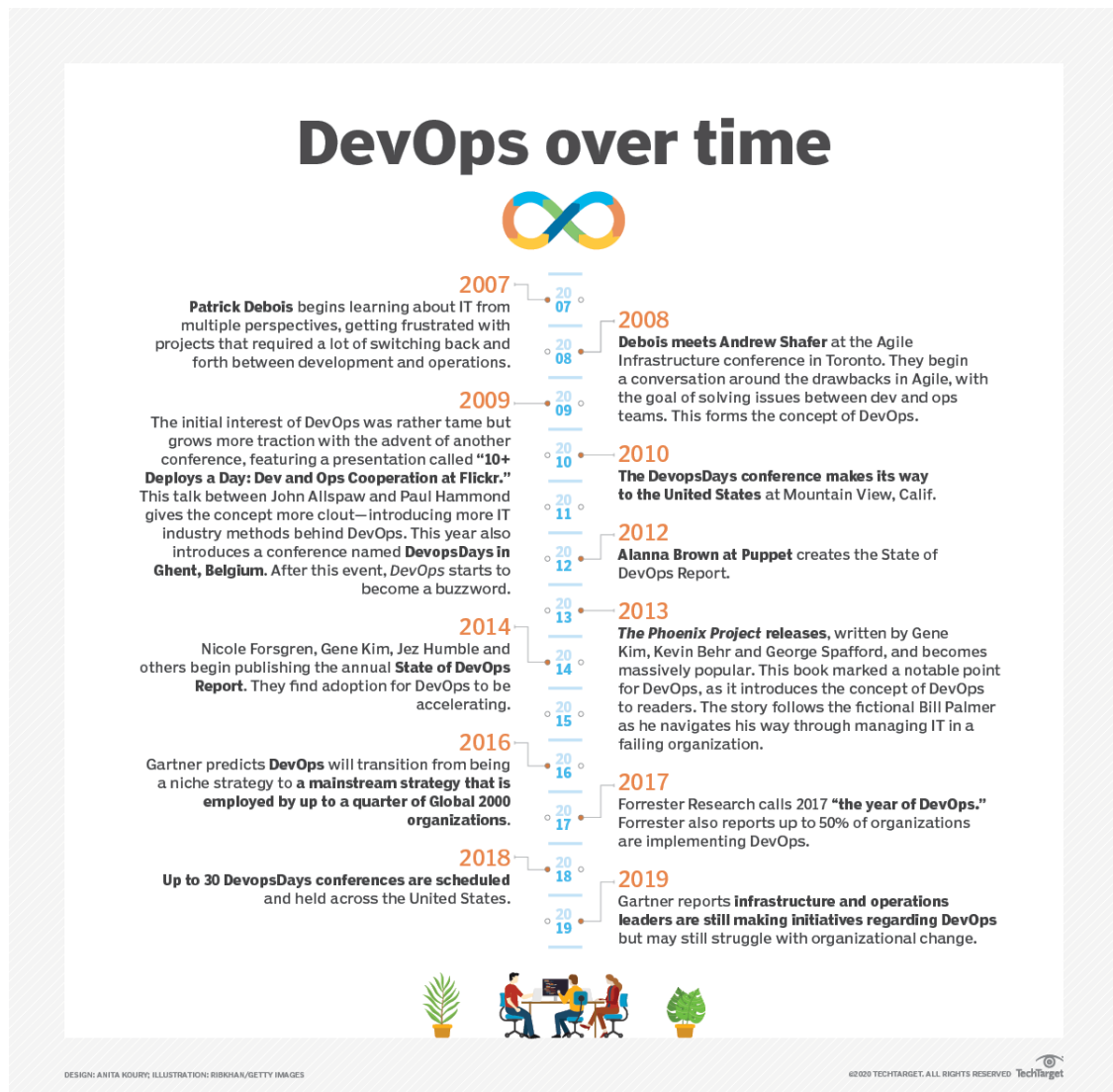


Figure 1: A timeline of the history of DevOps [11]

communications and cooperation between the development and operations teams. According to Allspaw and Hammond, a culture shift was needed in the industry to strengthen the bond between the different teams and the two convinced the software world that this was a key to success. Patrick Debois could not attend the conference physically, but joined the session remotely. As a response to this, Paul Nasrat encouraged Debois to organize his own event in Belgium, where he is from [19] [14].

2.3 DevopsDays conference

Later in 2009, the first conference on DevOps, called *DevopsDays*, was held in Ghent, Belgium. The arranger was none other than Patrick Debois himself. The word DevOps quickly became a buzzword after this conference and the next year the same conference was held in Mountain View, California. The US's adoption of the conference was a testament to the high interest in DevOps at the time. In 2018, as many as 30 DevopsDays conferences were held in different places in the US which shows the enormous growth of DevOps in that 9 year period [11] [19].

2.4 The Continuous Deployment Movement

Continuous deployment is the concept of continuously deploying each update to the codebase to the production environment. This idea was introduced by Timothy Fitz, who described it in a blog post back in February of 2009. Continuous deployment aims to be a way of rapidly identifying changes that break the production environment. Due to the high frequency of deployments it becomes easier to identify which change broke the system and also more convenient to fix it, as the change is fresh in memory. The quicker an error is noticed, the less damage it can do. The idea of continuous deployment quickly became a core concept of the DevOps pipeline and therefore this blog post has left a mark on the history of DevOps [10] [1].

2.5 The Phoenix Project

In 2013, Gene Kim, George Spafford and Kevin Behr wrote the book *The Phoenix Project*. The book tells the fictional story of how the main character, Bill Palmer, manages to save his company from its downfall through the implementation of DevOps principles. After its release this book rose to popularity and therefore also popularized the concept of DevOps further. The book was even used by professionals as a guide of how to implement DevOps in practice. For this reason The Phoenix Project is considered to be the DevOps "bible" by some [2] [23].

2.6 The State of DevOps Report

In the years 2012-2019 a yearly report on the state of DevOps was published by Puppet. The initiative was started in 2012 by Alanna Brown. In 2014, some of the biggest names in the scene; Nicole Forsgren, Gene Kim, Jez Humble began to publish the report [11].

2.7 The DevOps Handbook

Following the book *The Phoenix Project*, Gene Kim, Patrick Debois, Jez Humble and John Willis wrote the *The DevOps Handbook* in 2016. The book provides an extensive guide on how to implement the DevOps culture and principles and is written by the most influential people in the area [1].

3 Current best DevOps practices

It has been more than 10 years since the concept of DevOps was first introduced to the software industry. The practices have matured and DevOps is now standard among many companies in the industry. According to Diaz et al, it appears that the companies which incorporate DevOps practices into their workflows are more prone to succeed in competing on the market since DevOps allows for flexibility and adaptability which is very sought after in an ever changing market [8]. In the article *DevOps*, written by Ebert et al., the authors present the DevOps toolset, which consists of Build tools, Continuous-Integration, Operations, Deployment tools, Logging and Monitoring tools [9]. The following concepts summarize the current state of DevOps and can be considered best practices within the software world.

3.1 Version Control System (VCS)

A version control system (VCS) is a system that tracks changes made to files and simplifies the composition and management of different versions of software. A VCS is used to continuously develop, track changes and enable collaboration between developers when working on large systems. A CI/CD pipeline (see below) is integrated in the version control system such that the pipeline is run at each commit [17].

3.2 Continuous Integration (CI)

CI solves the problem that is merging code from a large number of developers working on the same codebase at the same time. By committing multiple times per day and running automated tests and builds on every commit it is easy to merge different contributions and find the "breaking" commit in case of a failure. The high frequency of commits also avoids situations where a local branch has diverged too much from the main branch and the merging of said branches becomes highly difficult [21].

3.3 Continuous Testing

Automated tests are run on *only* those parts of the system affected by a certain commit. This allows for tests to be run on each commit even on a large system where running every single test would take too long. The tests consist of unit tests, integration tests, performance tests and end-to-end tests. [13]

3.4 Continuous Delivery/Deployment (CD)

CD consists of two parts, Continuous Delivery and Continuous Deployment. Together these two processes make sure that each successful commit is continuously deployed to the production environment and also continuously released to the customers [20].

3.5 Logging and Monitoring

Logging and monitoring tools are used extensively in DevOps practices [9]. Logs of each commit, build and test are kept and managed. This way, if an error occurs, you have extensive documentation of the state of the system and history of actions that lead to that error. Also, logs help to catch larger errors before they arise [7]. Monitoring tools keep track of things such as "CPU load, RAM

allocation, network traffic statistics, memory consumption, and availability of free disk space.”. This way, the health of a system is continuously checked [9].

3.6 DevOps Culture

DevOps is not only about using tools, it is also about a shift in culture. As described by Gokarna & Singh, ”When embracing a DevOps culture, developers must take a fullstack developer approach, in which they take responsibility for the testing and release environment”. This implies that developers need to be trained in not only coding, but also testing. More than anything, the development and operations teams need to work closely and collaborate as if they are the same team [12].

3.7 Agile project management

Agile is a software development and project management concept that enables teams to deliver faster and according to the customers needs. The idea is to deliver small chunks of software frequently and continuously evaluate the product versus the needs and expectations of the customer. The dynamic and iterative approach of agile allows for constant evolution of a software project. It also enables goal-alignment between developers, operations and customers [25] [4].

4 Future of DevOps

Even though many problems within large-scale software development have been solved, there still is room for improvement. The software industry is dynamic and new practices and tools are introduced frequently. The following is a list of concepts that are starting to make their way into DevOps and will most likely shape the future of software.

4.1 AIOps & MLOps

A current trend that is making its way into software development and DevOps is the usage of Artificial Intelligence and Machine Learning to generate advanced analytics to boost the performance of the operations team [22]. Furthermore, it is likely that we will see the growth of AI-developer tools, such as GitHub Copilot, to accelerate the writing of code and tests [18].

4.2 Serverless

The concept of Serverless Computing involves deploying systems on external servers provided by vendors that are paid on an as-used basis. The main driving force behind switching to Serverless is to avoid the massive costs of having a server infrastructure and the risk of maintenance [6] [24].

4.3 Chaos Engineering

Chaos Engineering is a practice that involves experimenting on a system to improve confidence in the system’s stability. By subjecting a system to chaotic behaviours, such as crashes and severed network connections, one can simulate critical events that can happen in a production environment. This can uncover weaknesses in a system that are otherwise not found by standard testing practices [5] [15].

4.4 DevSecOps

DevSecOps is an extension of DevOps which aims to include the ever-growing important aspect of Computer Science that is security into software development. The idea is to incorporate security at all times during development of an artifact as opposed to adding security to an already finished artifact. Considering how important security is nowadays with the society transforming more into digitalization, it makes sense that more focus will be shifted towards security. Therefore, the standard will most likely shift from DevOps to DevSecOps. [28] [16].

5 Conclusion and reflection

In conclusion, DevOps is a concept that aims to bridge the gap between the development and operations teams in a software company. The concept was born out of the deficits in the Agile Method and was developed through a series of conferences, books and discussions. From being an idea discussed by two people at a conference in 2008, DevOps is now such a big concept that 30 conferences on the topic were held only in the US during 2018. Today, DevOps is vastly used throughout the industry and its practices are considered to be state of the art. Although the concept of DevOps has evolved a lot since the discussions between Patrick Debois and Andrew Shafer in 2008 the industry keeps changing and adapting each year. In the future, AI will augment more workflows and many companies will most likely move further towards a serverless infrastructure. Furthermore, a focus shift towards security is to be expected in the coming years.

When doing research on this topic, it became apparent to us that the history of DevOps is complex and it took many separate events to convince the industry of its perks. In general, it seems to take quite some time for people to adapt to new ideas. Ideas start out as buzzwords that only materialize after many years. Although many software developers are now practicing DevOps, it appears that it is hard to define what exactly the DevOps principles entails. We believe that in the future this will become a more well defined concept.

Take-home message: DevOps is a concept that has grown from an idea, born out of the deficits of agile, to being a cutting-edge methodology, that encompasses a plethora of practices, used by many in the industry.

References

- [1] Gene Kim. Jez Humble. et al. *The DevOps Handbook. How to Create World Class Agility, Reliability & Security in Technology Organizations*. IT Revolution, 2016.
- [2] Gene Kim. Kevin Behr. et al. *The Phoenix Project. A Novel about IT, DevOps, and Helping Your Business Win*. IT Revolution, 2013.
- [3] Atlassian. *DevOps principles*. URL: <https://www.atlassian.com/devops/what-is-devops>.
- [4] Atlassian. *What is Agile?* URL: <https://www.atlassian.com/agile>.
- [5] Principles of Chaos. *Principles of Chaos Engineering*. Mar. 2019. URL: <https://principlesofchaos.org>.
- [6] Cloudflare. *What is serverless computing?* URL: <https://www.cloudflare.com/en-gb/learning/serverless/what-is-serverless/>.
- [7] Devo. *Logging in a DevOps environment: what you should know*. Apr. 2019. URL: <https://www.devo.com/logging-in-a-devops-environment-what-you-should-know/>.

- [8] Jessica Diaz et al. “Why are many business instilling a DevOps culture into their organization?” In: *CoRR* abs/2005.10388 (2020). arXiv: 2005.10388. URL: <https://arxiv.org/abs/2005.10388>.
- [9] Christof Ebert et al. “DevOps”. In: *IEEE Software* 33.3 (2016), pp. 94–100. DOI: 10.1109/MS.2016.68.
- [10] Timothy Fitz. *Continuous Deployment*. Feb. 2009. URL: <http://timothyfitz.com/2009/02/08/continuous-deployment/>.
- [11] Alexander S. Gillis. *The history of DevOps: A visual timeline*. July 2020. URL: <https://www.techtarget.com/whatis/reference/The-history-of-DevOps-A-visual-timeline>.
- [12] Mayank Gokarna and Raju Singh. “DevOps: A Historical Review and Future Works”. In: *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*. 2021, pp. 366–371. DOI: 10.1109/ICCCIS51004.2021.9397235.
- [13] Tom Hall. *DevOps Best Practices*. URL: <https://www.atlassian.com/devops/what-is-devops/devops-best-practices>.
- [14] John Ellspaw. Paul Hammond. *10+ Deploys Per Day: Dev and Ops Cooperation at Flickr*. May 2009. URL: <https://www.slideshare.net/jallspaw/10-deploys-per-day-dev-and-ops-cooperation-at-flickr>.
- [15] Ori Hoch. *A Look Into DevOps Future – Top DevOps Trends?* July 2021. URL: <https://www.incredibuild.com/blog/a-look-into-devops-future-top-devops-trends>.
- [16] Marc Hornbeek. *Best of 2021 – Nine Pillars of DevOps Best Practices*. Dec. 2021. URL: <https://devops.com/nine-pillars-of-devops-best-practices/>.
- [17] JetBrains. *What is Version Control?* URL: <https://www.jetbrains.com/teamcity/ci-cd-guide/concepts/version-control/>.
- [18] Gaurav Kumar. *What Does GitHub’s Copilot Mean for the Future of the Software Industry?* Oct. 2021. URL: <https://www.bairesdev.com/blog/what-githubs-copilot-mean-for-the-future/>.
- [19] Frederic Paul. *The Incredible True Story of How DevOps Got Its Name*. May 2014. URL: <https://newrelic.com/blog/nerd-life/devops-name>.
- [20] Sten Pitett. *Continuous integration vs. delivery vs. deployment*. URL: <https://www.atlassian.com/continuous-delivery/principles/continuous-integration-vs-delivery-vs-deployment>.
- [21] Max Rehkopf. *What is continuous integration?* URL: <https://www.atlassian.com/continuous-delivery/continuous-integration>.
- [22] ScienceLogic. *What is AIOps?* URL: <https://sciencelogic.com/solutions/aiops>.
- [23] TechTarget. *The Phoenix Project*. Sept. 2020. URL: <https://www.techtarget.com/whatis/definition/The-Phoenix-Project>.
- [24] Veritis. *Future of DevOps: Top 6 DevOps Trends in 2022 and Beyond*. URL: <https://www.veritis.com/blog/future-of-devops-top-6-devops-trends-in-2022-and-beyond/>.
- [25] Wikipedia. *Agile software development*. URL: https://en.wikipedia.org/wiki/Agile_software_development.
- [26] Wikipedia. *DevOps*. URL: <https://en.wikipedia.org/wiki/DevOps>.

- [27] Wikipedia. *Systems development life cycle*. URL: https://en.wikipedia.org/wiki/Systems_development_life_cycle.
- [28] Kev Zettler. *DevSecOps Tools*. URL: <https://www.atlassian.com/devops/devops-tools/devsecops-tools>.