The Impact of DevOps on Cloud Development

Filip Bäck (filbac@kth.se)

May 2022

1 Introduction

The working methodology within software development flourished once the agile methodology were introduced to the sector. The more dynamic way of working entailed shorter working periods that allowed for more flexibility within the development process. Once the agile methodology were cemented as common practice, its dynamic processes became applicable to other areas within the sector and was adopted as what today is called DevOps. Cloud computing is tightly intertwined with the DevOps practices that are common today, and enables many of the features that make DevOps so prevalent. While DevOps has benefited greatly from the adoption of cloud services, it is also beneficial the other way around. The growth of DevOps through out the tech industry has put higher requirements on the cloud platforms, which has enabled them to grow and become standard practice within the industry.

2 The Evolution of Software Development Methods

The concept of computers dates all the way back to the 17th century, and has been in constant development ever since [10]. Although it was only a theoretical practice at the time, it set the standard for the machine we use today. The general purpose computers were not put in practice until the 1950s which defines the birth of software development, by programming in low level languages [15]. Later in the 20th century the development of computer had picked up pace and large corporations in the technical sectors started the development of software systems and products on a larger scale [6]. While this type of evolution within the industry was seen as a success, it quickly emerged problems with how to organize the work when multiple developers worked on the same projects [20].

The organisational problems were partly solved with the introduction of version control tools that simplified the management of the different software versions. Additionally, the way of working within software development were in hindsight less than optimal, as the main method of working was the famous waterfall method [12]. One of the pitfalls of the waterfall method within software development was that the development was divided into phases, where each phase was dependant on the previous one. This caused problems as software projects famously are often hard to define beforehand and thus making it hard to plan for, as well as being able to visualize how the end result would turn out [14]. This eventually gave birth to the more dynamic way of working, the agile development method 1. The agile method builds on short periods of work with the focus of having a working project after each period [21]. This combined with close contact with the customer/project owner enables the involved parties to discuss the current product and how to progress during the next period of work. The agile method quickly became a success and is widely accepted by software developers globally [4].

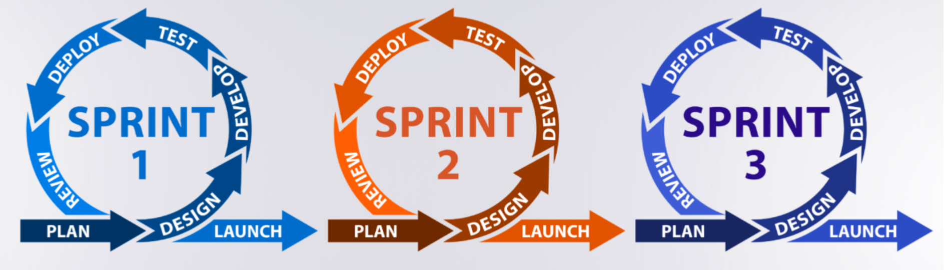


Figure 1: A visualisation of the agile method

3 The birth of DevOps

While the adoption of the agile way of working increased the efficiency of teams working within software development, there was a gap in the communication with their counterpart in the IT operations team [3]. The roles within a corporation working with software development traditionally consisted of developers that created a software application with documentation and ship it to a QA team, which in turn installs and tests the application. If the application would pass certain criteria it would be handed off to the production operations team. This team would then be responsible for the deployment and management of the software without much communication with the developers [17]. This way of working was eerily similar to the waterfall method, where the process is static and the individual teams are highly dependant on the processes of each other. It also made it difficult to make changes to the application once it was sent down the pipeline [19].

The endeavor of making the development-to-production pipeline more efficient were crucial for the software development industry, and in the early 2010s it gave birth to what is today called DevOps [1]. DevOps originates from the combination of development and operations, with the goal of making the production pipeline of software applications more efficient. The principles of DevOps originates from the agile method of working, where the focus lies on being dynamic and always having a stable build that can be improved upon. The main principles of DevOps can be summarized by the following terms [1].

- Continuous Integration and Continuous Delivery (CI/CD) - Teams can roll out changes to a software application in a safe, systematic way which minimizes the risk of breaking the application. This is enabled by the use of tools which help keep track of system states.

- Version Control - The practice of managing machine code in versions. This enables the team to revise and remove/edit code in a safe way, via e.g. Git.

- Agile Software Development - A way of working within software development that emphasizes cooperation and high adaptability. Enables a dynamic working environment in response to feedback from users/customers/project owners.

- Infrastructure as Code - Defines resources in a manner that allows for the management of those resources as that of code.

- Configuration Management - The management of resources in a system, e.g. databases, VMs, and servers.

- Continuous Monitoring - Having real-time information about the health and performance of the application stack. This helps teams mitigate possible errors during production.

As such, DevOps can be seen as the adoption of agile methods. The development of applications within the software industry is taking a more dynamic approach instead of the traditional monolithic one. While DevOps includes specific tools that enables this type of work it includes more changes to the workspace, such as cultural impact, possibility for automation, and flexible file sharing [11].

4 The Evolution of Cloud Computing

The idea of cloud computing dates back to the 1960s when tech companies like IBM and DEC started to dabble in the concept of time-sharing, i.e. splitting computing resources among many users simultaneously. The concept were further evolved during the 90s by telecommunications companies that offered VPN services, as they could lower their costs by optimizing bandwidth usage by redirecting traffic to balance server use. It was not until around 2010 when the cloud platforms that are prevalent today, e.g. AWS, GCP, and Azure, was shipped to market[9].

Cloud computing comes in various forms as of today with the most common product offerings being infrastructure[1], platform[2], software[3], and serverless computing[4]. These services are delivered through the internet making cloud based services available everywhere as long as there is a network connection. The benefits of cloud platforms exceeds the ability to store and access data remotely, as it also allows for outsourcing of both services and hardware. Companies does not need to build, structure and maintain their own tech infrastructure as it is now available online. This can potentially save resources such as time, money, and personnel in addition to making the services available remotely as well[18].

The cloud services are also highly flexible in regards to updates and patches; if the cloud provider ships updated to their services the user will get access to it right away. Additionally, the cloud services are highly scalable. The IT resources from a cloud provider can be adjusted as needed in order to meet the users demand, making it efficient for both parties. While utilizing cloud computing can be beneficial for users, it can also pose significant risks. The outsourcing of IT infrastructure makes the user susceptible for the risks of their cloud provider. Storing sensitive data externally may make for a potential security threat as the user loses power over how it is handled. The cloud provider can also face hardware problems, bug issues, etc., making the user vulnerable to disadvantages they can not control[7].

[1] IaaS - Infrastructure as a Service

[2] PaaS - Platform as a Service

[3] SaaS - Software as a Service

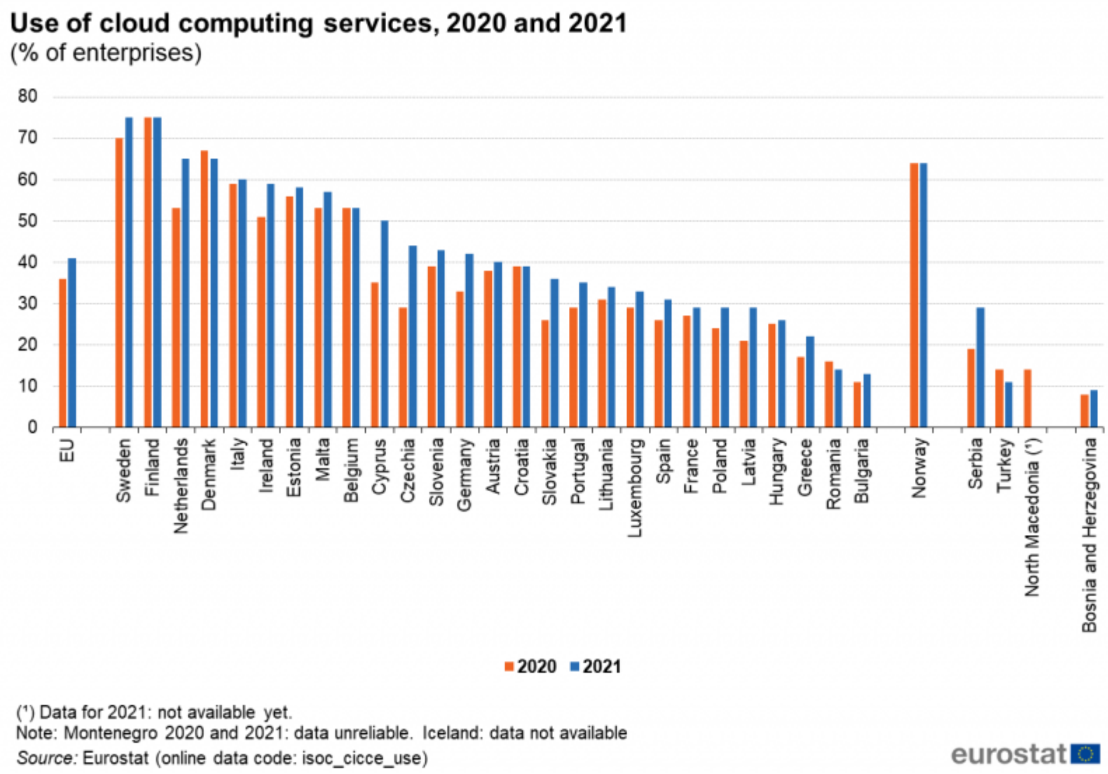
[4] FaaS - Function as a Service

5 How DevOps Impacts Cloud Development

The collaboration between the developers and the IT operations team that is DevOps has evolved to being more than tools that benefit faster software delivery, it has reshaped the process. The addition of cloud computing to the DevOps development process has come to define how entire industries work [16]. The approach has went from monolithic to agile, with testing being automated, enhanced control over the product life cycle, better collaboration, centralized architecture, and increased speed in the workflow [11]. The automation of agile methodologies gives the developers total control of the production cycle of applications, enabling them to respond to requirements in real time [2].

While cloud computing has been detrimental to the development of the DevOps practice, DevOps has been as important to the development of the cloud services. This mutually beneficial relationship has impacted the cloud services in more ways than one. While the DevOps practice has helped enable the cloud platforms by offering reliable services that fit it well, it has also been embraced within cloud development [11]. A relevant factor within cloud services is being able to meet their users demand in a timely manner, where the improved tracking of the usage of resources can help enable this and improve the service offering. The automation of processes that comes with the DevOps practice has in addition to improving the work cycle of applications become an essential part of cloud platforms today, as they can offer that practice to users via their platform [13]. As automated CI/CD pipelines has become an essential part of the software development process, it provides great value to both the user as a service as well as the cloud services as a service offering. The automated pipelines enables the shipping of quality features to market at a faster pace than previously, making it possible for businesses to gain a competitive advantage in their industries, which holds true for both the cloud developers and their potential users [2]. As organisations put more emphasis on DevOps practices, it might highlight a need for different resources to satisfy different parts of the organisation which can't be fulfilled with DevOps only. Cloud development has become such a vital part of the development process that it can encourage organisations to develop their own cloud service to speed up the process [8]. The DevOps practice might dictate the new methods, but it does not have much merit without the Cloud.

A study conducted by Eurostat (European Statistical Office) of EU member countries showed that countries with a developed technical sector such as Finland, Sweden, and Denmark had roughly 70% of its enterprises that utilizes cloud computing services, with increasing numbers each year [5]. Additionally, the average number of enterprises trough out the EU member states that uses cloud services was around 41%. While it is stated that most of the usage from the cloud services were used for e-mail and storage of files, the usage and growth of cloud services might not be as prevalent as it is without the recent development of DevOps practices.


Use of cloud computing services, 2020 and 2021

(% of enterprises)

(¹) Data for 2021: not available yet.

Note: Montenegro 2020 and 2021: data unreliable. Iceland: data not available

Source: Eurostat (online data code: isoc_cicce_use)

Figure 2: Graph showing the use of cloud computing services

References

[1] Microsoft Azure. *What is DevOps?* 2022. URL: https://azure.microsoft.com/en-us/overview/what-is-devops/#practices (visited on 05/23/2022).

[2] Tech Blogger. *DevOps and Cloud: A Symbiotic Relationship*. 2020. URL: https://contenteratechspace.com/blog/devops-and-cloud-a-symbiotic-relationship/ (visited on 05/21/2022).

[3] Meredith Courtemanche, Emily Mell, and Alexander S.Gillis. *What Is Agile Methodology? - Overview Of Agile Software Development And Agile Models*. 2021. URL: https://www.techtarget.com/searchitoperations/definition/DevOps (visited on 05/20/2022).

[4] Digite. *What Is Agile Methodology? - Overview Of Agile Software Development And Agile Models*. 2022. URL: https://www.digite.com/agile/agile-methodology/ (visited on 05/22/2022).

[5] Eurostat. "Cloud computing - statistics on the use by enterprises". In: *European Statistical Office* (2021). ISSN: ISSN 2443-8219.

[6] History Extra. *A brave new world: the 1980s home computer boom*. 2016. URL: https://www.historyextra.com/period/20th-century/a-brave-new-world-the-1980s-home-computer-boom/ (visited on 05/18/2022).

[7] Jake Frankenfield. *Cloud Computing*. 2021. URL: https://www.investopedia.com/terms/c/cloud-computing.asp (visited on 05/21/2022).

[8] KSolves. *How DevOps Is Dictating A New Approach to Cloud Development*. 2021. URL: https://www.ksolves.com/blog/devops/how-devops-is-dictating-a-new-approach-to-cloud-development (visited on 05/23/2022).

[9] Reto Meier. *An Annotated History of Google's Cloud Platform*. 2017. URL: https://medium.com/@retomeier/an-annotated-history-of-googles-cloud-platform-90b90f948920 (visited on 05/22/2022).

[10] CHM - Computer History Museum. *The Babbage Engine*. 2022. URL: https://www.computerhistory.org/babbage/ (visited on 05/17/2022).

[11] Dharmalingam N. *DevOps and Cloud Computing - A Winning Combination*. 2022. URL: https://www.whizlabs.com/blog/devops-and-cloud-computing/ (visited on 05/23/2022).

[12] Kai Petersen, Claes Wohlin, and Dejan Baca. "The Waterfall Model in Large-Scale Development". In: *Product-Focused Software Process Improvement*. Ed. by Frank Bomarius et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 386–400. ISBN: 978-3-642-02152-7.

[13] Google Cloud Platform. *CI/CD on Google Cloud*. 2022. URL: https://cloud.google.com/docs/ci-cd (visited on 05/23/2022).

[14] Tutorial's point. *SDLC - Waterfall Model.* 2022. URL: `https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm` (visited on 05/19/2022).

[15] J R.Harris. *The First Modern Computer – The Case for Baby, the Manchester Mk I Prototype.* 2003. URL: `http://virtualtravelog.net/2003/08/the-first-modern-computer-the-case-for-baby-the-manchester-mk-i-prototype/` (visited on 05/17/2022).

[16] M Rajkumar et al. "DevOps culture and its impact on cloud delivery and software development". In: *2016 International Conference on Advances in Computing, Communication, Automation (ICACCA) (Spring).* 2016, pp. 1–6. DOI: `10.1109/ICACCA.2016.7578902`.

[17] New Relic. *What Is DevOps.* 2022. URL: `https://newrelic.com/devops/what-is-devops` (visited on 05/20/2022).

[18] Sam. *DevOps and Cloud: How DevOps is Dictating a New Approach to Cloud Development.* 2021. URL: `https://huddle.eurostarsoftwaretesting.com/devops-and-cloud-how-devops-is-dictating-a-new-approach-to-cloud-development/` (visited on 05/22/2022).

[19] Veritis Transcend. *Waterfall Vs. Agile Vs. DevOps - Which Production Method Should You Take?* 2022. URL: `https://www.veritis.com/blog/waterfall-vs-agile-vs-devops-which-production-method-should-you-take/` (visited on 05/21/2022).

[20] Mikah Yost. *A Brief History of Software Development.* 2018. URL: `https://medium.com/@micahyost/a-brief-history-of-software-development-f67a6e6ddae0` (visited on 05/19/2022).

[21] Shi Zhong, Chen Liping, and Chen Tian-en. "Agile planning and development methods". In: *2011 3rd International Conference on Computer Research and Development.* Vol. 1. 2011, pp. 488–491. DOI: `10.1109/ICCRD.2011.5764064`.