
Adopting Continuous delivery - The common challenges developers face, their causes and solutions

Author: *Benjamin Ronneling*

Date: *May 2, 2022*



Contents

1	Introduction	3
2	Background	3
2.1	Continuous Integration (CI)	4
2.2	Continuous Delivery (CD)	4
2.3	Continuous Deployment	5
2.4	Why do organizations consider CI/CD?	5
3	Method	6
4	Results	6
4.1	Challenge 1: Gaining support from different stakeholders	7
4.2	Challenge 2: CD adoption is complicated and moves forward slowly	8
4.3	Challenge 3: Getting sustained support over a long period of time	8
4.4	Challenge 4: Development teams lack a common vision	8
5	Discussion	9
6	Conclusion	9

1. Introduction

Software development has evolved into a complex task as tools and infrastructure of software development have grown. A strategy for combatting these complexities and meeting the demand for faster software development is to automate parts of the development, testing, and release of the software. This can be done by adopting Continuous Integration (CI), Continuous Delivery (CD), and Continuous Deployment which are also known as CI/CD pipeline. [1]

Adopting CI/CD pipeline is a long process and depending on the company size, it can take up to 4 years or more to transform from a traditional agile methodology to DevOps with CI/CD pipeline [2]. This process can even get longer if the DevOps transition team is not familiar with the transformation challenges. Thus, in this essay, the goal has been to present some of these challenges in order to answer the following essay questions:

What are some of the common challenges that developers face when they adopt Continuous Delivery in their companies? What are the causes and solutions to these challenges?

In the background section of this essay, a description of the traditional development process and CI/CD pipeline is presented. In order to answer the essay questions, a literature review has been conducted where 4 challenges were identified and are presented in the results section. Each challenge is described together with its cause, and a solution to each challenge is suggested. In the discussion section, the limitations of these results are discussed. Finally, a conclusion is presented in the last section.

2. Background

The DevOps transition started around 2007 when software development and IT operations noticed that there was a dysfunction in the industry. The traditional software development model consisted of the software development team and the IT operation team and these teams often had separate objectives and often worked in different buildings or floors [3]. Most of the developers used agile methodologies for planning and development which had short cycles that included planning, development, testing, and deployment as shown in figure 1. The agile methods remain mainstream in modern software development because they provide product adaptivity to the always-changing market and customer needs [4].

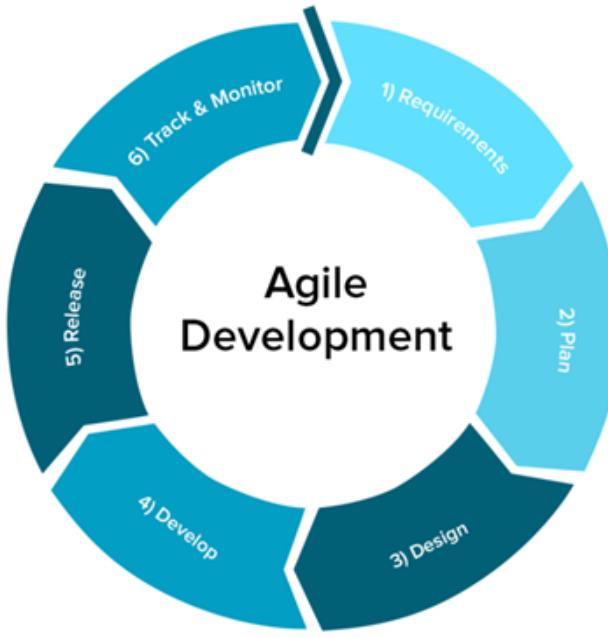


Figure 1: An iteration of agile development which takes 1-2 weeks [5].

The problem with the traditional development methods is that they required engineers to manually check the code to ensure code quality and to update changes in the codebase. These manual actions result in weeks-long or even month-long delays. Traditional agile also has trouble bringing the developers who write the software, together with the operations team who runs the software in production and builds and maintains the infrastructure where the code is executed. [3]

DevOps is the solution to the traditional development methods which replaces the old approach where the development team wrote the applications and then threw them over the wall to an operations team that had minimal knowledge of how the code was developed [4]. DevOps automates development, testing, and releasing of the software using three phases: Continuous Integration, Continuous Delivery, and Continuous Deployment. [6]

2.1. Continuous Integration (CI)

Continuous Integration is the practice of automating the integration of code changes from multiple developers into a single software project. CI allows developers to frequently merge their code changes into a central repository where it builds and then tests. A version control system keeps track of new changes and makes it possible to revert to an older version in case tests are not passed. [7]

2.2. Continuous Delivery (CD)

Continuous delivery is an extension to CI that automatically deploys all code changes to a testing environment after the build stage. The main idea of CD is to have any update

ready for release at any time. This is different from agile methods where iterations take 1-2 weeks, and the new features must wait before being released to production. [5][8]

2.3. Continuous Deployment

Continuous Deployment further automates the CI/CD pipeline. With Continuous Deployment, every change in the code is automatically pushed to production. Figure 2 shows how the “deploy to production” stage in Continuous Delivery is done manually with respect to Continuous Deployment which automatizes this stage. [1]

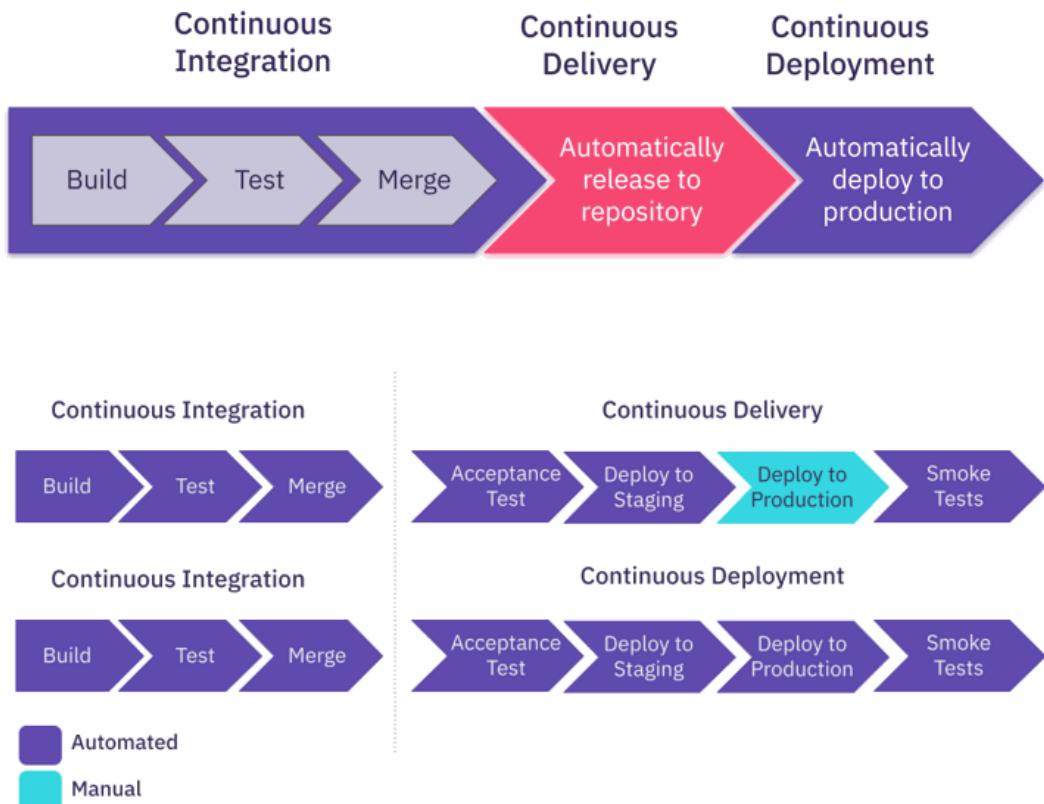


Figure 2: Difference between Continuous Integration, Delivery, and Deploy [1].

2.4. Why do organizations consider CI/CD?

Continuous Delivery increases the velocity of development and decreases the time gap between proposing a new feature and its delivery. The number of possible mistakes and bugs is significantly reduced as every update undergoes verification stages [5]. CI/CD also enhances teamwork, and the automation allows for continuous and fast collaboration between software developers, quality engineers, and the operations team [1]. Many organizations are willing to start integrating CI/CD into their development process because of these benefits.

Adopting Continuous Integration can be a challenging process depending on the experience of the team that makes the transition. Regardless, having knowledge of the different

challenges results in a faster and a successful transition. Therefore, some of these challenges are presented in the results section which can be useful when adopting CD. In the next section, a short description of the methods used to find these challenges are presented.

3. Method

A literature review has been done to answer the essay questions in this essay. The literature review consisted of studying related research articles in bibliographic databases and websites. A case study of a team that has adopted CD practices into the development process of their company, has also been reviewed that contains data from real-life CD transition. A master thesis article which consists of interviews with the stakeholders after CD adoption is also reviewed to get a glimpse of the cultural challenges that stakeholders face. By studying different research papers, 4 challenges with solutions were identified and they are discussed in the results section.

“Continuous Delivery: Overcoming adoption challenges” is the paper that Lianping Chen has written about his experience on CI/CD adoption in his company. The described transition in the article has been done in Paddy Power which is a betting company with approximately 5000 employees. The CD transition team in Paddy Power consisted of 8 people which worked 4 years to complete the transition [2]. Studying this paper gave an insight into how a large company like Paddy Power might experience the adoption of CI/CD and what cultural issues the employees might face. The author of this paper has included some suggestions for the challenges they faced which are presented in the results section.

4. Results

Through reviewing the literature, 4 challenges to CD transition were identified which are presented in the table below. Each challenge is described in the following subsections together with its cause. A suggestion about how to overcome the challenge has also been provided for each challenge at the end of each subsection.

CHALLENGE	CAUSE	SUGGESTED SOLUTION
Gaining support from different stakeholders	Stakeholders are often under time pressure and have own goals that they prioritize	Identify the stakeholders' issues and describe the solutions that CD provides for them
CD adoption is complicated and slow	CD adoption is done in part time	Dedicate multidisciplinary CD transition team
Getting sustained support over a long period of time	restructuring of the organization	Deliver CD continuously
Development teams lack a common vision	There is not enough guidance for the new CD practice	Dedicate a DevOps Coach

Table 1: A summary of identified challenges, causes and solutions

4.1. Challenge 1: Gaining support from different stakeholders

Many different areas of an organization need to be changed in order to adopt CD. These changes include automatization of build, test, and deployment processes, change in the organization's culture, training in new software development practices, and so on. Support from a wide range of stakeholders including the infrastructure team, the top management, the system engineering, security, and the operation team is needed in order to make such substantial changes. Gaining support from all these groups is a huge challenge because all these groups have their own everyday work and goals that they prioritize. For example, in the case study of Paddy Power, the software development team was constantly under time pressure and had very busy schedules. Gaining their support meant that they had to work overtime for their own tasks which was challenging for them. The security team had the same time pressure while the releases happened weakly. For the security team, CD adoption meant that the number of releases would increase from weakly to daily releases which would create yet more time pressure on them, and thus it was even harder to gain their support. [2]

A solution to this challenge is to find the problems that each stakeholder has which CD adoption can solve. Lianping Chen calls this strategy "selling CD as a painkiller". For example, the transition team in Paddy Power identified that the software development team spends on average 20 percent of their time setting up or fixing their testing environment. They sometimes had to wait for weeks in order to obtain the testing environment from other teams for their new projects. CD could solve this problem by setting up the testing environment and running the tests automatically in minutes. When this was explained to the software development team, they were happy to schedule the needed time to make changes on their side. The same solution was applied to the security team where they did not have enough engineers to run the security checks frequently and CD could solve this issue by running the security checks automatically. They also became interested in collaborating after this solution was explained to them. This shows that identifying the stakeholders' issues and describing the solutions that CD provides for them is a good

strategy to gain their support. [2]

4.2. Challenge 2: CD adoption is complicated and moves forward slowly

CD adoption is a time-consuming task that needs a lot of effort. CD adoption also needs different skills to transform different sections of the organization, from development to the security section. A cause for the slow adoption of CD can be that there is no dedicated team that takes care of the transition. Making engineers work on CD transition part-time can slow down the transition as they can easily be pulled off the transition task and work on the other tasks. Another reason for the slow adoption can be that the transition team does not have the right competency for transitioning all the different applications. [2]

A solution to this challenge is to have a dedicated team of developers with members from different development backgrounds. This way, the team can focus on the transition only without the need to worry about other tasks. Having members with different backgrounds from different departments helps a lot when the team faces a complicated task that needs special skills. [2]

4.3. Challenge 3: Getting sustained support over a long period of time

As described in the first challenge, getting support from different stakeholders is a challenging task. Getting this support from the stakeholders over a long period of time can be even more challenging. One reason for this can be the restructuring of the organization and changes in the leadership of the company. For example, in the case study of Paddy Power, under the 4 years of transition to CD, the CEO and the CTO of the company had changed. At the same time, the transition team's over manager had changed five times and the organization had been restructured three times. Every time these changes occurred, the implementation value of CD was questioned. If the benefits of CD adoption are not visible to the new managers, there is a risk of them dropping their support for the transition plan. [2]

A solution to this challenge is to continuously deliver CD. This means that the easiest (but most important) part of the adoption must be done first so that the effectiveness and suitability of the adoption are shown to the stakeholders. This delivers the value of CD early on to the company and helps developers to align their practices with CD gradually. It also gives the developers a chance to give feedback on the adoption process and the new tools. [2]

4.4. Challenge 4: Development teams lack a common vision

When CD is adopted, the development teams are responsible for following CD practices and using the CD pipelines. Different development departments in the organization might apply the CD practices differently which results in lack of a common vision. According to the interviews made by Gustav Lindström in his thesis, developers found themselves in a state of confusion and uncertainty, especially during the initial stages of the CD adoption, as there is not enough guidance for the developers that are unfamiliar with the

new practices. These developers had to create their own plans on how to apply the CD practices which resulted in lack of a common vision. [9]

A solution to this problem is to establish common ways of working and spread the knowledge through a DevOps Coach. A DevOps Coach can be a person, team, or a central unit within the organization that defines the organization's interpretation of DevOps. The organizations should allocate time and resources for the developers to learn the different elements of DevOps and their new responsibilities. [9][10]

5. Discussion

The discussed challenges in the result section are based on the experience of the transition team of Paddy Power and the interviews that Gustav Lindström has written about in his essay. Both articles point out important challenges and solutions that can be an enormous help for the companies that want to adopt CD. These articles described many other challenges that are not included in this essay, and it is a good idea to study these articles fully before starting CD adoption.

Another interesting article that describes the more technical challenges that the developers face, during CD adoption, is "Problems, causes and solutions when adopting continuous delivery—A systematic literature review" by Eero Laukkanen, Juha Itkonen, Casper Lassenius. One example of the technical challenge in this article is the time-consuming testing, where the developers had to wait extremely long for the CI servers to test the code. The article points out the importance of having powerful hardware resources to run the tests in order to shorten the testing time, which should be considered from the beginning when the transition to CD starts. [11]

6. Conclusion

The key take-away of this essay is that CD adoption is a challenging process that needs dedicated resources and time, and knowing some of these challenges helps the transition process go smoothly and makes the transition time shorter. The challenges described in this essay are "gaining initial and long-term support from different stakeholders", "slow CD adoption" and "the lack of common CD practices". To overcome these challenges, there should be a dedicated multidisciplinary CD transition team that makes the transition happen. The transition team needs to gain the support of the different stakeholders by pointing out the solutions that CD provides to the stakeholders' everyday problems. They need to focus on the easiest adoption parts of CD first, to make the results visible to the stakeholders in the early stages and get their feedback [2]. Providing a DevOps coach helps to create a common vision of CD practices for the developer and prevents them from being in a state of confusion [9].

References

- [1] Alastair Wilkes. *A Crash Course on Continuous Delivery / Launchable, Inc.* Launchableinc.com, Feb. 2022. URL: <https://www.launchableinc.com/a-crash-course-on-continuous-delivery/#What%20%99s-the-difference-between-Continuous-Delivery-and-Continuous-Deployment?> (visited on 05/02/2022).
- [2] Lianping Chen. *Continuous Delivery: Overcoming Adoption Challenges*. ResearchGate, Feb. 2017. URL: https://www.researchgate.net/publication/313874428_Continuous_Delivery_Overcoming_Adoption_Challenges (visited on 05/02/2022).
- [3] Atlassian. *History of DevOps / Atlassian*. Atlassian, 2022. URL: <https://www.atlassian.com/devops/what-is-devops/history-of-devops> (visited on 05/02/2022).
- [4] Atlassian. *Agile vs DevOps / Atlassian*. Atlassian, 2016. URL: <https://www.atlassian.com/devops/what-is-devops/agile-vs-devops> (visited on 05/02/2022).
- [5] *Continuous Delivery and Integration: Rapid Updates by Automating Quality Assurance*. AltexSoft, 2017. URL: <https://www.altexsoft.com/blog/business/continuous-delivery-and-integration-rapid-updates-by-automating-quality-assurance/> (visited on 05/02/2022).
- [6] *CI/CD concepts / GitLab*. Gitlab.com, 2020. URL: <https://docs.gitlab.com/ee/ci/introduction/> (visited on 05/02/2022).
- [7] *What is Continuous Integration? – Amazon Web Services*. Amazon Web Services, Inc., 2022. URL: <https://aws.amazon.com/devops/continuous-integration/#:~:text=Continuous%20integration%20is%20a%20DevOps,builds%20and%20tests%20are%20run.> (visited on 05/02/2022).
- [8] Continuous integration | Thoughtworks. *Continuous integration / Thoughtworks*. Thoughtworks.com, 2022. URL: <https://www.thoughtworks.com/continuous-integration> (visited on 05/02/2022).
- [9] Gustav Lindström. *The Challenges of adopting DevOps*. DIVA, 2019. URL: <http://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A1372216&dswid=-7455> (visited on 05/02/2022).
- [10] JULIE PETERSON. *CI/CD and the Promise of Agile Transformation / WhiteSource*. WhiteSource, Apr. 2022. URL: <https://www.whitesourcesoftware.com/resources/blog/all-about-ci-cd/> (visited on 05/02/2022).
- [11] Eero Laukkanen, Juha Itkonen, and Casper Lassenius. “Problems, causes and solutions when adopting continuous delivery—A systematic literature review”. In: *Information and Software Technology* 82 (Feb. 2017), pp. 55–79. doi: 10.1016/j.infsof.2016.10.001. URL: <https://www.sciencedirect.com/science/article/pii/S0950584916302324> (visited on 05/02/2022).