

**EXAMENSARBETE** Continuous Delivery: Challenges, Best Practices, and Important Metrics

**STUDENTER** Anders Klint, Vilhelm Åkerström

**HANDLEDARE** Lars Bendix (LTH), Axel Franke (Bosch), Peter Walls (Bosch)

**EXAMINATOR** Elizabeth Bjarnason (LTH)

# Challenges, Best Practices, and Important metrics for Continuous Delivery of Software

---

POPULAR SCIENCE SUMMARY **Anders Klint, Vilhelm Åkerström**

---

Software teams being able to frequently and painlessly deliver new updates of their software to their customers has many benefits, however accomplishing this is not always easy. We have explored challenges and best practises of this type of workflow, as well as which metrics are valuable when evaluating the effectiveness of the process.

Releasing software can often be a tedious process requiring many work hours and quality checks required for a satisfactory result. Continuous Delivery (CD) is a software development workflow which aims to eliminate the tediousness of this process, and which has become very popular in recent years. To achieve this, the goal of CD is to automate steps such as building, testing, and releasing the software such that quality software can be released with a push of a button at any time in the development process.

However, with CD still being a new way of working there is a lot of uncertainty of how to do it well. The change to CD from other software development methods can not be done in just a day or a week. This is a gradual process which faces many different challenges. We performed a study to discover what these challenges are, in addition to some best practices to aim for when looking to achieve this workflow.

Some of the common challenges were keeping up with automation, due to technical challenges such as tests taking too long to run, or a lack of commitment to automate from developers or managers wanting to focus on developing new features. In

order to combat this, it is important for everyone involved to be on board with the processes and work necessary to do CD, in addition to sharing responsibility for the CD-pipeline, as well as making sure that the team has proper experience of CD through training and hands-on experience.

Additionally, some metrics gathered from literature were used to monitor a team having achieved CD at the case company. This resulted in three major findings: Firstly, that limited licenses to software tools used to quality check newly written code can throttle the process if the number of instances of the software is limited by the number of licenses. Secondly, that a tool used to generate documentation on each code change took longer to run than anticipated, and could thus be run less frequently. Thirdly, that when measuring duration, it is important to not only measure the time taken for each process to run, but also the idle time between processes.

These findings of the monitored team could inspire other teams working with CD of how to improve their workflow. Additionally, the challenges, best practices, and important metrics found could be useful for practitioners and future researchers.