CI/CD

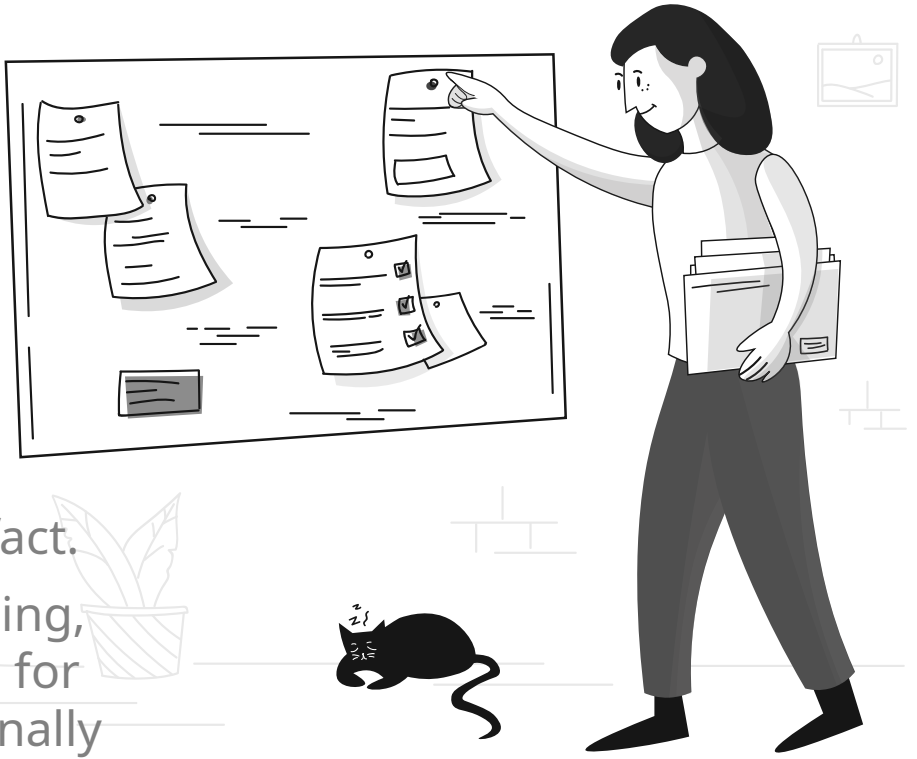
The correct way to build and ship

Tobias Hobmeier

23.11.2021

Continuous Integration

- Continuous Integration
  merging all developers' commits
  to a shared mainline
  - done on a frequent base will avoid
  conflicts in the future. It's the first step
  to deliver a high quality, and working artifact.
- Steps in this stage include: compiling, testing,
  running static analysis as well as checking for
  vulnerabilities in the our dependencies. Finally
  the code artifacts are stored



Illustrations by Pixeltrue on icons8

Continuous Deployment

> This is the process by which verified changes in codebase or system architecture are deployed to production as soon as they are ready and without human input.
> For doing so we set up infrastructure, provisioning servers, copying files, smoke testing, promoting to production and even rolling back a change if something did not look right.

Benefits of CI/CD

Automated Smoke Tests

Finding bugs right away will protect our revenue. And will avoid downtime caused by deploy-related crash or bugs

Faster and More Frequent Production Deployment:

We get more revenue by shipping value generating features more frequently to the customers, this would also help us to get feedback early and stay ahead.

Deploy to Production Without Manual Checks:

Less time to market will help us to make more money

Catch Unit Test Failures:

Having less bugs in our deployment and spending less time doing manual debugging would help us to avoid cost

Detect Security Vulnerabilities:

This would enable us to easily detect serious security flaws that would be embarrassing if it had made it to the public. This would save us money trying to win back the customers' trust and rebuilding our image