

**Introductie Continuous Integration** was het vijfde onderdeel van de leergang Future Proof Testing. Docent Marthijs trok ons de wereld in van pipelines, versiecontrole, continuous delivery, devops, deploys, de testpyramide en commits.

## Jobs en pipelines

Veel ontwikkelorganisaties zeggen aan CI te doen, maar hoe goed gaat dat in de praktijk? In de training leerden we dat versiecontrole een belangrijk fundament is van softwareontwikkeling. Met versiecontrole zorgen ontwikkelaars dat steeds de juiste versie van de code wordt geïntegreerd. Als nieuwe of gewijzigde code wordt ingecheckt in het VCS (versiecontrolesysteem) gaat het balletje rollen: bouw van een compleet softwareproduct, statische checks, geautomatiseerde tests en uitrol (deployment) naar verschillende omgevingen volgen, steeds vaker volledig automatisch. De verschillende taken worden 'jobs' genoemd die samen een 'CI pipeline' vormen.

## Jenkins

Marthijs nam ons mee Jenkins in, een veel gebruikte tool om de jobs in een CI-pipeline te zetten en uit te voeren. Jenkins heeft de regie over het bouwproces, de statische checks die op de code worden uitgevoerd, de dynamische geautomatiseerde tests en de uitrol op verschillende omgevingen. Eventueel volgen meer geautomatiseerde tests op test- en acceptatieomgevingen en zelfs uitrol naar productie.

## Een job is rood

Als het allemaal goed gaat dan is 'alles groen'. Maar wat als er iets 'rood' is? Dan begint de analyse. We oefenden in het bekijken van de verschillende dashboards en moesten uitvinden welk probleem er was in verschillende casussen, zoals:

- falende tests
- te hoge complexiteit van de code
- te lage coverage van de unit tests
- code die niet aan coding standaarden voldoet

## Artifacts

Bij artifacts denk ik nog vaak aan archeologische opgravingen. Maar bij software artifacts is het niet de bedoeling dat die bedolven raken... Belangrijk artifact voor testers is de meest recente stabiele versie van het product om op te testen. Waar het VCS zich richt op de broncode, draagt een artifact repository zorg voor het zorgvuldig beheren van executables, librabries en third-party software.

## De tester in CI

Voor de tester is het van belang om het CI-proces te overzien en te beoordelen op de mate waarop de juiste checks en tests in plaatsvinden. De tester interpreteert de output van die checks en tests en vindt zo belangrijke informatie over de kwaliteit van het product onder test.

En zo hebben we weer een mooi pakketje kennis opgedaan in de leergang!

**Kees Blokland**