

De rol van testautomatisering in CI/CD

In de [vorige blog](#) hebben we gekeken naar de eisen die testautomatisering stelt aan testomgeving en testdata. In deze blog staan we kort stil bij CI/CD en de rol die testautomatisering hierbij speelt.

CI/CD

Testautomatisering is een voorwaarde voor **continuous integration (CI)**, **continuous deployment (CD)** en **continuous delivery (CD)**. Bij CI en CD is continu inzicht vereist in de kwaliteit van de code en het product terwijl deze continu aan verandering onderhevig zijn. Dit is onmogelijk zonder het gebruik van efficiënte en effectieve testautomatisering.

-> Continuous integration

Bij continuous integration is het belangrijk om de kwaliteit van nieuwe en/of aangepaste code op een code branch zoveel mogelijk te borgen voordat de nieuwe en/of aangepaste code samengevoegd wordt met de bestaande code. Omdat het verstandig is vaak te integreren, moeten de geautomatiseerde tests op de branch snel uitgevoerd kunnen worden. In de testpiramide hebben we gezien dat de tests onder in de piramide snel zijn. Unit- en componenttests zijn dus de aangewezen tests om toe te passen op een ontwikkelbranch. Het al dan niet slagen van de unit- en componenttests wordt bij CI gebruikt als quality gate voor het mogen mergen van een branch naar de master.

-> Continuous deployment

Bij continuous deployment wordt de code automatisch uitgerold naar één of meerdere testomgevingen voor het uitvoeren van functionele testen, ketentesten, acceptatietesten of het testen van een ander kwaliteitsattribuut. In de testpiramide hebben we gezien dat deze tests wel te automatiseren zijn maar dat deze in het algemeen trager zijn in de uitvoering. Naarmate de omvang van het product groeit zal ook de hoeveelheid uit te voeren tests en daarmee de testdoorlooptijd groter worden. Als de doorlooptijd van het uitvoeren van alle tests te groot wordt voor de gewenste doorlooptijd van de pipeline, is het verstandig om de tests te gaan managen. Een manier hiervoor is het maken van subsets. Door tests onder te brengen in subsets naar bijvoorbeeld functionaliteit of subketen, kan risicogebaseerd een subset van de totale tests worden uitgevoerd om gedurende de dag de pipeline snel te houden. Gedurende de nacht kan dan de totale testset draaien om de kwaliteit van het product met voldoende diepgang te blijven testen.

Ook is het verstandig om bij continuous deployment subsets te definiëren die gebruikt kunnen worden als smoketest voor het controleren van de automatisch gegenereerde testomgevingen voordat de test op die omgeving wordt afgetrapt.

-> Continuous delivery

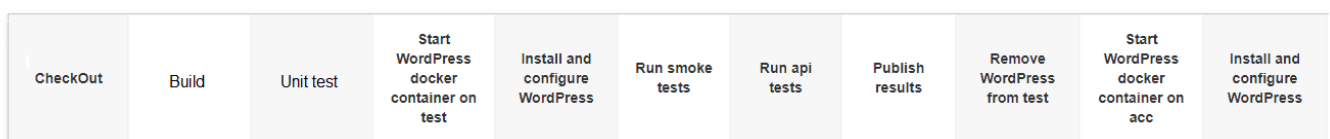
Bij continuous delivery is ook de laatste stap, het naar productie brengen van code, geautomatiseerd. Waar bij continuous deployment bepaalde tests in de pipeline nog met de hand uitgevoerd worden, zijn bij continuous delivery alle testen in de pipeline geautomatiseerd, inclusief de acceptatietesten en een geautomatiseerde smoketest op productie om te beoordelen of de uitrol goed is verlopen.

Omdat in continuous delivery geen plaats meer is voor handmatig, exploratory testen in de pipeline vraagt het een hoge volwassenheid van alle fasen in het proces. Productrisico's die we in een niet continuous delivery setting met exploratory tests afdekken moeten we nu op een andere manier ondervangen. Continuous delivery gaat dus samen met risico mitigerende maatregelen als:

- Canary releases (geschaald naar productie gaan).
- A/B tests in productie.
- Intensieve productiemonitoring op alle levels (server, applicatie en business monitoring).

Genoemde mechanismen bieden de mogelijkheid om productrisico's in productie te mitigeren. Met canary releases en A/B tests kan het exploratory testen in productie plaatsvinden. Eventuele problemen in productie worden snel gevonden (monitoring), de impact van de problemen blijft beperkt (canary releases, A/B tests) en kunnen snel opgelost worden (snelle pipeline).

Onderstaande figuur laat een gedeelte van een CI/CD-pipeline zien met de daarin aanwezige stappen (klik op afbeelding voor groter beeld):



Vooruitblik volgende blog

Tot zover deze blog. In de volgende en laatste blog kijken we naar het **groeipad** bij de invoering van testautomatisering.

STAY TUNED!

Wim ten Tusscher