

Technical Test Specialist Jesse Huisman gaf tijdens de afgelopen Polteq Conferentie de presentatie “BDD beyond use case testing”. Hieronder de samenvatting van zijn presentatie.

‘Behaviour Driven Development’ ofwel ‘**BDD**’, het is een van die buzzwoorden die we allemaal weleens tegenkomen in het veld. Het lijkt een radicaal nieuwe visie op testen te geven, maar wat niet iedereen weet, is dat BDD sterk geïnspireerd is op een klassieke testtechniek genaamd ‘use case testing’.

De meer bekende voorloper van BDD is ‘Test Driven Development’ oftewel TDD. TDD ontstond als een development methode waarbij met de slogan ‘test first’ een radicale ‘shift left’ toegepast kon worden op de laagste test levels. Door al voorafgaand aan het ontwikkelen een unittest te schrijven, werd het mogelijk om direct testdekking in te bouwen in de native codetaal.

Waar TDD als development methode ophield, pakte BDD als testaanpak het stokje over. Het ‘test first’ principe zou binnen BDD toegepast kunnen worden op de hogere test levels. Het format van de use case test werd nu omgevormd naar de Given-When-Then syntax van de pseudo code ‘Gherkin’. Dit zorgde voor een voor iedereen leesbare notatie die door business opgesteld kon worden en zich ook goed leende voor (test)automatisering.

BDD en testautomatisering

Het is de testautomatisering die BDD echt tot zijn recht laat komen. BDD Tools als ‘Cucumber’ en ‘SpecFlow’ maken het mogelijk om de opgestelde ‘Feature Files’ als tests te automatiseren. De Given-When-Then stappen zijn hierbij te parameteriseren met keywords wat het mogelijk maakt in één keer meerdere test cases te dekken. Ook is het mogelijk om ‘living documentation’ te creëren door deze tests geregeld te draaien en de resultaten te publiceren.

Helaas is niet alles rozengeur en maneschijn. Als de business de feature files opstelt, dragen ze dan ook zorg voor de testdekking? Kun je dit überhaupt hebben als je vastzit aan die ene testtechniek ‘use case testing’ en is het wel mogelijk om in een ‘agile’ setting al voorafgaand aan de implementatie je tests op te stellen?

Met het ‘3 amigos’ idee, dat suggereert dat bij het opstellen van de feature files niet alleen een vertegenwoordiger van de business, maar ook een tester en een developer betrokken moeten zijn, geeft BDD eigenlijk zelf al het antwoord op de laatste vraag. Het ‘test first’ principe blijft echter een core principe van BDD. Dit principe lijkt niet te rijmen met onze huidige ‘agile’ werkwijzen, waar documentatie onvolledig en veranderlijk kan zijn en ‘exploratory testing’ een ‘must’ is. Dan rest de vraag: ‘Kun je volstaan met die ene testtechniek: use case testing?’

Drie pointers

Naar mijn mening kun je je als tester prima staande houden in een BDD setting zolang je een set belangrijke pointers niet uit het oog verliest:

1. Claim je rol als 1 van de 3 amigo's & breng je test expertise in;
2. Laat je niet dwingen om 'test first' toe te passen maar test 'just on time'
3. Gebruik 'use case testing' maar laat je er niet door beperken en exploiteer je BDD tooling om gedegen testdekking te realiseren

Jesse Huisman, Technical Test Specialist bij Polteq

Wil je meer weten en leren over Behaviour Driven Development? In september en november wordt bij Polteq de tweedaagse training [Praktisch Behaviour Driven Development](#) (met Java/Cucumber en/of C# / SecFlow) gegeven. Inschrijven kan via onze website.