

# Groeiende vraag naar technische testers

Door Jeroen Mengerink



De wereld verandert en dat brengt andere eisen aan testers met zich mee. Vooral het technische aspect van testen komt steeds meer naar voren. De snelle opkomst van cloud computing zorgt ervoor dat veel functionaliteit uit de cloud kan worden afgenomen. Naast het testen bij de business of deze functionaliteit de processen goed ondersteunt, is ook technisch testen nodig. Om de functionaliteit te integreren met bestaande systemen, zal namelijk een technische koppeling nodig zijn. Daarnaast zal voor het monitoren van de correcte werking in productie testautomatisering ingericht moeten worden, waar ook weer technische testers voor nodig zijn.

**D**e andere grote “hype” die meespeelt bij de vraag naar technische kennis is Agile. Door het inzetten van multidisciplinaire teams, komt de tester dichterbij de ontwikkelaar aan te zitten. De communicatie tussen deze partijen, die voorheen vooral via uitgebreide documentatie liep, zal nu direct plaatsvinden. Om deze communicatie soepel te laten verlopen, zullen de ontwikkelaars wat van testen moeten weten, maar zullen de testers dus ook kennis van ontwikkelen nodig hebben. Het iteratieve en incrementele karakter dat Agile heeft, vraagt ook om het steeds vaker uitvoeren van een regressietest. De noodzaak voor testautomatisering neemt hierdoor ook toe.

In dit artikel beschrijf ik voor cloud, Agile en testautomatisering enkele situaties die aangeven waar de vraag naar technische testers vandaan komt

### Cloud

Meer en meer bedrijven nemen diensten af uit de cloud, denk hierbij aan e-mail, CRM of zelfs hele omgevingen. Al deze diensten worden via het internet benaderd en moeten geïntegreerd worden in de huidige bedrijfsprocessen. Het testen of de communicatie met deze cloud services goed gaat is een technische bezigheid. Er zal goed gekeken moeten worden naar het berichtenverkeer, wat zonder GUI vaak beter bekeken kan worden.

Het gebruik van cloud services heeft ook grote invloed op interne ontwikkeltrajecten. Wanneer ketens getest moeten worden vanwege interne aanpassingen, is het veelal ongewenst om dit te doen terwijl de koppeling met de (live) cloud service actief is. In deze gevallen zal dus een stub of mock gemaakt moeten worden, zodat de cloud service gesimuleerd kan worden en de processen getest kunnen worden zonder de productieversie van de cloud service te raken.

In het geval van afspraken of contracten met de leverancier over de prestaties van de cloud service zal in productie gekeken moeten worden of de leverancier zich wel aan zijn afspraken houdt. Er ontstaat dus vraag naar frequente (of zelfs continue) tests die het gedrag van de service kunnen monitoren.

Hiervoor is goede testautomatisering nodig, waar ik later in dit artikel op terugkom.

### Agile

Bij Agile wordt uitgegaan van vier simpele basisprincipes:

- Mensen en hun onderlinge interactie boven processen en tools
- Werkende software boven allesomvattende documentatie
- Samenwerking met de klant boven contract-onderhandelingen
- Inspelen op verandering boven het volgen van een plan

Dat onderlinge interactie belangrijk is, is te merken in de multidisciplinaire teams. De minder uitgebreide documentatie vereist dat de mensen in de teams goed met elkaar communiceren over wat ze aan het doen zijn. Om te kunnen begrijpen wat andere teamleden doen, zal enige kennis van de aanwezige vakgebieden nodig zijn. Vandaar dat van de testers verwacht wordt dat zij in ieder geval basis programmeerkennis en -vaardigheden bezitten. Het is belangrijk om mee te kunnen discussiëren over software architectuur en mogelijke oplossingen. De oplossingsrichting kan namelijk door de teams bepaald worden.

## Het is belangrijk om mee te kunnen discussiëren

Door het kort-cyclische karakter zullen ook vaak tests uitgevoerd worden op producten die nog niet volledig af zijn. Hiervoor zal de tester vaak technische voorzieningen in moeten zetten. Denk, net als bij cloud, aan het gebruik van drivers, stubs of mocks. Het maken of inrichten van deze testondersteunende voorzieningen zal door de testers aangegeven (en bij voorkeur ook gerealiseerd) moeten worden.

In korte iteraties moet werkende software opgeleverd worden, die voortborduurde op opleveringen in eerdere iteraties. Elke iteratie komt er nieuwe functionaliteit bij, die in de volgende iteratie nog steeds moet werken. Ofwel, de regressietestset groeit gestaag en de tijd om de

tests uit te voeren is beperkt. Het regelmatig uit moeten voeren, samen met de tijdsdruk, levert een goede business case voor testautomatisering. Het goed nadenken over en inzetten van testautomatisering zijn vereisten voor een goed Agile traject.

Gezien de regelmatige opleveringen is het belangrijk om goed versie- en configuratie-management te hebben. Ook de testers zullen hier aan moeten bijdragen. Wanneer en hoe moet geïntegreerd worden en welke eisen worden aan de verschillende ontwikkelomgevingen gesteld? Technische kennis zal hier helpen om een beter en duidelijker beeld te krijgen van wat zal moeten gebeuren.

### **Testautomatisering**

Dat aan testautomatisering technische aspecten zitten is niet te ontkennen. Aangezien zowel bij cloud als bij Agile testautomatisering naar voren komt, is voldoende aanleiding om hier specifieke aandacht aan te besteden. Zeker gezien het feit dat ook in andere trajecten steeds meer vraag is naar testautomatisering.

Testautomatisering kan plaatsvinden op verschillende niveaus. De basis wordt gelegd in unit testen. Hoewel de unit tests over het algemeen door de ontwikkelaars gemaakt worden, betekent dit niet dat we hier als testers niets mee te maken hebben. De ontwikkelaars kunnen geholpen worden door het toepassen van structurele technieken, waar de testers over het algemeen meer kennis van hebben. Door middel van pair programming kan een tester hier ondersteuning bieden, door tijdens het maken van unit tests naast de programmeur te zitten om zo aan te geven welke testgevallen gemaakt moeten worden. Naast het helpen bij het opstellen, kunnen de unit tests natuurlijk ook gereviewd worden. Ook hier is weer kennis van en vaardigheden met programmeren nodig.

Functionele testautomatisering ligt meer bij de testers. In de Agile trajecten is te zien dat de grafische interface regelmatig aangepast wordt, om de automatisering robuust te houden zal deze moeten plaatsvinden op de laag eronder. Het aanroepen van functies zonder de grafische interface te gebruiken, vereist kennis van de manier waarop geprogrammeerd is. Wanneer toch automatisering gemaakt wordt die de grafische



interface gebruikt, zal deze op enige manier aangeroepen moeten worden. Hoewel het weinig technisch werk lijkt doordat bijvoorbeeld Fitnesse [1] of Cucumber [2] gebruikt kan worden door de testers, zal toch enig programmeerwerk nodig zijn. Het aanspreken van het testobject kan namelijk alleen maar wanneer voor Fitnesse en Cucumber een interpretatielaag aanwezig is. Meestal voldoen de standaard meegeleverde mogelijkheden niet en moet aanvullende functionaliteit geprogrammeerd worden. Vaak zal van de tester verwacht worden dat ook deze laag ingericht wordt.



In meerdere contexten is vraag naar continue testen. Bij Agile om te zien of regressie optreedt naar aanleiding van nieuwe opleveringen, bij cloud om te monitoren of de leverancier zich aan zijn afspraken houdt, etc. De enige manier om continue (of in ieder geval frequente) testen goed uit te voeren, is door deze te automatiseren. Het nadenken over hoe dit te realiseren en het daadwerkelijk inrichten vraagt om kennis van architectuur, programmeren en testen.

### Conclusie

De software die gemaakt wordt is steeds complexer. Dit vraagt niet alleen om meer technische kennis van ontwikkelaars, maar ook van testers. Van testers wordt steeds vaker verwacht dat zij kunnen programmeren (bij voorkeur in verschillende programmeertalen) en dat ze mee kunnen denken en discussiëren over de architectuur van de software. Bij het gebruik van de cloud is integratie met verschillende systemen nodig, waarbij het kunnen lezen van technische logging en het gebruik van drivers, stubs en mocks van de tester geëist wordt. Doordat bij Agile de tester dicht tegen de ontwikkelaar aanzit, zal ook de communicatie op een technischer niveau plaatsvinden.

Daarnaast moet de tester blijven in de snel veranderende wereld, waar zowel Agile als cloud steeds vaker de standaard is. De enige manier om niet achter te raken, is het inzetten van testautomatisering. Hiervoor zullen ondersteunende tools niet alleen gebruikt, maar ook begrepen en ingericht moeten worden. Niet vreemd dus dat de vraag naar technische testers groeit.

Blijf dus leren en verbreed je kennis, dan kunnen wij als test community voldoen aan deze vraag!

[1] Meer info over Fitnesse kun je vinden op <http://fitnesse.org/>

[2] Meer info over Cucumber kun je vinden op <https://github.com/cucumber/cucumber/wiki/>



Jeroen Mengerink werkt sinds 2008 bij Polteq en is testconsultant. Naast zijn werkzaamheden voor klanten is hij betrokken bij diverse testinnovaties. Jeroen is het eerste aanspreekpunt bij Agile-vraagstukken voor collega's en klanten. Hij is docent van een gevarieerd pakket aan testtrainingen, onder meer met onderwerpen als Agile, SOA en Cloud. Daarnaast ligt zijn persoonlijke interesse op het gebied van testautomatisering.

Hij is een van de auteurs van het boek Cloutest® over het testen van cloudservices. Ook is Jeroen regelmatig te zien als spreker op conferenties zoals onder andere Testnet, EuroStar en ChinaTest.

