

Interview met Martijn de Vrieze

Gepubliceerd 10 juli 2013 | Door [Rob van Steenberg](#)
martijn.devrieze@polteq.com • [@martijndevrieze](#)

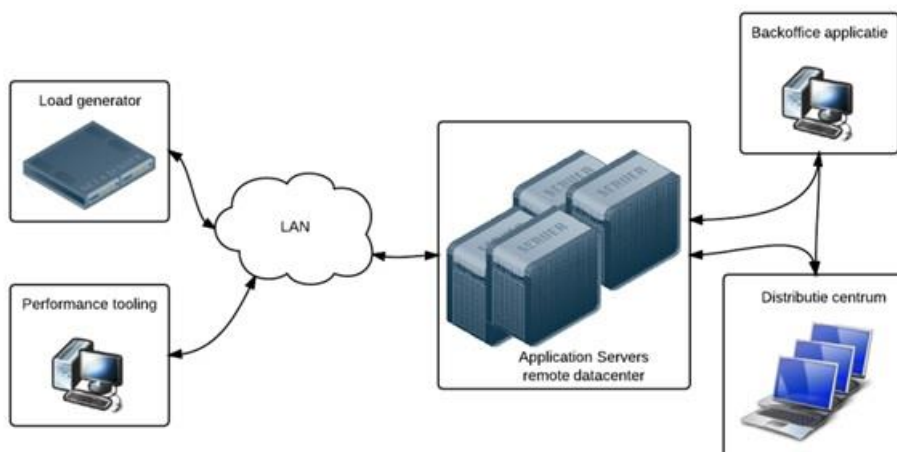


Deze week een interview met Martijn de Vrieze, testconsultant bij Polteq Test Services en op dit moment werkzaam bij Jumbo Supermarkten als performancetester.

Ik zie op LinkedIn dat je op dit moment vooral bezig bent met automatisch testen en specifiek performancetesten. Kan je ongeveer aangeven wat je doet op dit vlak in je huidige opdracht?

Ik ben in mijn huidige opdracht inderdaad steeds meer in performancetesten. Sterker nog, het is een van de voornaamste zaken waarop ik geselecteerd ben. In het verleden heb ik bij bedrijven met uitermate high-volume sites als Marktplaats en Spil Games gewerkt en bij beide ben ik in aanraking gekomen met load- en performancetesten.

Binnen mijn huidige opdracht (in de Retail branche) wordt er hard gewerkt aan de integratie van twee afzonderlijke platforms. Daar komt erg veel zeer technisch testen bij kijken. Een reden hiervoor is dat het aantal gebruikers, het aantal werknemers en winkels met deze integratie verdubbelt ten opzichte van het huidige gebruik. De gehele end-to-end keten, van inkoop tot verkoop, eigenlijk tot aan het distributiecentrum, komt daarmee onder druk te staan. Om dit alles te kunnen accommoderen worden er fundamentele wijzigingen aan de infrastructuur en programmatuur doorgevoerd. Aan deze wijzigingen hangen sterke functionele requirements welke zich uiten in non-functionele tests, zoals de reactietijd van applicaties en functionaliteit binnen de applicaties.



Kortom, om een lang verhaal iets langer te maken, ik help de opdrachtgever met het opbouwen van load op de (nieuwe) servers en afhankelijk van de requirements van dat moment voer ik vervolgens metingen uit op de uiteindelijke performance. Oftewel reactietijd van de applicaties zoals de

eindgebruiker deze gebruikt. Hiervoor gebruik ik een aantal verschillende applicaties om onafhankelijk van elkaar load op te bouwen en de performance te meten.

Welke applicaties gebruik je om deze load- en performancetests uit te voeren?

Per applicatie en per test verschilt het welke tools ik gebruik of misbruik om load- en performancetests uit te voeren.

Tools die ik wel veel gebruik, zijn de diverse opensourcetools zoals Jmeter en The Grinder. Reden dat ik die tools vaak gebruik, is dat ze zeer breed en flexibel inzetbaar zijn. Zo kun je met Jmeter, als je wilt, niet alleen load opbouwen maar ook een functionele geautomatiseerde test opbouwen.

Tools die ik daarnaast veel gebruik zijn SmartBear's SoapUI en LoadUI, maar ook UltraEdit en NotePad++ misbruik ik erg veel, hoewel die laatste twee voornamelijk om in zeer rap tempo veel data tegelijkertijd te wijzigen en aanpassen aan mijn specifieke eisen en wensen.

Op dit moment gebruik ik ook een puur functionele testtool, Sikuli, om daarmee de reactie tijden van een applicatie te meten vanuit gebruikersoogpunt, terwijl ik via een loadgenerator als Jmeter functionele load genereer op de servers.

Testen in het algemeen wordt toch nog erg vaak onderschat bij bedrijven, zo is mijn ervaring. Heb jij een idee of dit ook zo geldt voor performancetests of zie je hier een andere trend?

Als ik kijk naar wat er in de Nederlandse markt gebeurt ten aanzien van load-, performance- en geautomatiseerd testen heb ik zeker het idee dat dit zwaar onderschat wordt in een groot deel van de markt. Vaak zijn de echte technologiegeoriënteerde bedrijven al wel bezig met wat testautomatisering en sommige zijn daarin verder dan andere. Maar van vooral de grotere bedrijven die erg veel baat kunnen hebben bij geautomatiseerd testen, doen er nog te weinig mee. De traditionele manier van testen in veel grotere organisaties, is een groep key-gebruikers uit hun reguliere werk halen en ze een paar weken mee te laten lopen om handmatig regressie tests uit te voeren (enigszins gechargeerd).

Wat veel bedrijven daarmee pogen te bereiken is een duidelijk beeld van wat de status is van de applicatie of het applicatielandschap na een upgrade op basis van 'vertrouwde' flows, uitgevoerd door mensen die dat voor hun dagelijks werk doen. Wat daarmee echter wordt gemist, is de gedachtegang van een tester. Kortom, niet alleen kijken naar de happy-flows of voorspelbare fout-flows, maar ook kijken naar alle omringende systemen en bedrijfsprocessen die geraakt worden.

Door, bijvoorbeeld, end-to-end tests en regressie tests te automatiseren verwijder je een stuk van de vaak lange doorlooptijd in deze testfasen. Daarnaast creëer je ruimte om variaties te gaan uitvoeren, dus de minder gebaande paden, die wel vaak problemen opleveren. Denk aan schrikkeljaren, jaarafsluitingen, jaarwisselingen en andere bijzondere condities die in een systeem kunnen optreden. Maar daarnaast ook extensievere tests die de omringende systemen raken om daarop vast te stellen of alles nog goed werkt.

Daarnaast zie ik een andere uitdaging met betrekking tot geautomatiseerd testen die niet zozeer bij bedrijven ligt, maar meer binnen het vakgebied. Veel 'traditionelere' testers vinden code zelf schrijven en onderhouden vaak eng of niet binnen hun taken pakket vallen. 'Ik ben een tester, geen ontwikkelaar', is iets wat ik vaak heb gehoord.

```

1 using System;
2 using System.Text;
3 using System.Text.RegularExpressions;
4 using System.Threading;
5 using NUnit.Framework;
6 using Selenium;
7
8 namespace ExampleSeleniumTests
9 {
10     [TestFixture]
11     public class ExampleSeleniumTest
12     {
13         private ISelenium selenium;
14         private StringBuilder verificationErrors;
15
16         [SetUp]
17         public void SetupTest()
18         {
19             selenium = new DefaultSelenium("localhost", 4444, "chrome", "http://www.w3schools.com/");
20             selenium.Start();
21             verificationErrors = new StringBuilder();
22         }
23
24         [TearDown]
25         public void TeardownTest()

```

Met de grote opkomst van agile testen en andere methoden waarmee sneller nieuwe functionaliteit wordt opgeleverd en in productie genomen, is het in mijn ogen juist zaak om zo snel mogelijk alle herhaaldelijke tests te automatiseren. Of zoals Michael Bolton ontzettend duidelijk heeft gemaakt in zijn blogs, 'checks' te automatiseren zodat je als tester volledig kunt focussen op dat stuk waar onze echte toegevoegde waarde ligt: het cognitieve proces van testen.

Als ik om me heen kijk, zie ik dat veel collega's in de diverse opdrachten het heel erg interessant vinden om te zien wat we allemaal kunnen bereiken met geautomatiseerd testen. Wanneer je vervolgens vraagt of zij het ook zouden willen of kunnen, is het antwoord vaak in de richting van 'nee, ik kan geen code schrijven', of nog erger 'nee, ik ben geen ontwikkelaar!'.

Het kan ook zijn dat testers vaak niet weten waar ze moeten beginnen. Als ik kijk naar het aanbod van test tools, dan schrik ik zelf ook wel eens. Moet ik nu Java gaan leren, Visual Basic of mij verdiepen in Cucumber achtige manieren ten behoeve van testautomatisering. Heb jij een tip hiervoor?

Om het antwoord heel simpel te stellen: daar is geen eenduidig antwoord op te geven. Het ligt er helemaal aan wat je voor jou werk op dat moment nodig hebt.

Voor welke vorm van test automatisering dan ook is het overigens altijd handig, vaak zelfs noodzakelijk, om een redelijke basiskennis van programmeren te hebben. Dus als je ergens wilt beginnen zou ik zeggen, duik in eerste instantie in een vorm van programmeren. Of dat nou Python of Java is maakt uiteindelijk niet enorm veel uit, tenzij je het doet om een specifieke tool goed te begrijpen. De basisbeginselen van programmeren zijn gewoon erg belangrijk omdat testautomatisering een vorm van programmeren is of in ieder geval bevat.

Er is een aantal goede plekken om te starten, commerciële trainingen zoals die door Polteq worden gegeven maar ook zelfstudieprogramma's als '[Code academy](http://codecademy.com/)' of 'learn **** the hard way' (bijvoorbeeld: <http://learnpythonthehardway.org/>)

Kortom, er is geen echt 'one size fits all' antwoord te geven over waar te beginnen; het hangt helemaal af van wat je behoefte is.

Ben je nog van plan om een presentatie te geven op een testevenement of is er al iets gepland?

Ik heb al diverse keren presentatie voorstellen ingestuurd voor diverse events, van Testnet tot EuroStar maar tot nu toe niet gekozen. De meeste voorstellen zijn direct gelieerd aan

testautomatisering, waarom het vaak faalt (of faalde), hoe je kunt zorgen dat het wel werkt en zich terugbetaalt et cetera. Eigenlijk vooral in de richting van [waar ik ook over blog](#).

Hoe zie je jezelf over vijf jaar? Ben je dan nog tester?

Over een jaar of vijf? Dan denk ik zeker dat ik nog altijd in het testvak zit. Het is iets waar ik jaren geleden in ben gerold, maar in al mijn nieuwe banen steeds verder aan verknocht geraakt ben. Er is nog genoeg om te leren en te zien. Om maar niet te beginnen over wat er nog allemaal aan nieuwe technologische uitdagingen aankomt. Meer applicaties die uit een cloudoplossing worden geserveerd, testautomatisering op thin clients, mobile apps en sites. En straks het concept van applicaties op technologieën als Google Glass. Kortom, er is nog zoveel leuks te ontdekken in de technologiewereld dat ik erg uitkijk naar wat er gaat veranderen in de komende vijf jaar.