

# Tooling kiezen vraagt actuele kennis van de testmarkt

De weg naar geautomatiseerde ondersteuning van de automatisering is inmiddels op de kaart gezet. Ook binnen het testvak. Er is een ruime keus aan testtools, zowel open als closed source. Welke tool het meest geschikt is, hangt af van de behoefte en daarmee ook van de vraag of het gaat om tooling voor de ontwikkeling van systemen of voor het beheer hiervan in de productiesituatie.

*Reinder Otter, Maurice Siteur en John van Veen*

Wie heeft er belang bij tools? Deze vraag staat centraal bij de aanschaf ervan. Als niemand een tool wil gebruiken, is er immers ook geen noodzaak voor een project, afdeling of organisatie om hem aan te schaffen. Het omgekeerde is overigens niet per definitie aan de orde: als er wel een behoefte is om een tool te gebruiken, moet vaak eerst een budgethouder overtuigd worden van de noodzaak ervan.

Bij dat proces is de noodzaak niet altijd duidelijk voor de beslisser. Zo zal de aanschaf van een ontwikkelplatform voor bijvoorbeeld een .Net- of Java-project geen issue zijn, want zonder zo'n platform kan een project geen software realiseren. Het gebruik van een specifieke debuggertool is minder vanzelfsprekend. De toekomstige gebruiker zal de besluitvorming moeten beïnvloeden om met concrete argumenten budget beschikbaar te krijgen

De aanschaf van tools die gebruikt worden bij testen is evenmin vanzelfsprekend. Een testtool is een geautomatiseerd hulpmiddel dat ondersteuning biedt aan één of meer testactiviteiten, zoals planning en beheer, specificatie, opbouwen uitgangbestanden, testuitvoering en beoordeling. Een testtool is pas een hulpmiddel is als het gebruik

ervan hogere productiviteit en/of efficiëntie oplevert; het mag geen doel op zich zijn om een tool te gebruiken. Vaak zien we dat bij de aanschaf van een testtool een relatief grote investering nodig is. Als de besluitvorming alleen wordt gebaseerd op de opbrengst van een tool binnen de ontwikkeling van een systeem, is het zeer de vraag of de aanschaf van bijvoorbeeld een tool waarmee men geautomatiseerde functionele tests kan uitvoeren, zal worden terugverdiend. De besluitvorming gaat dan voorbij aan de noodzaak van een geautomatiseerde regressietest bij de beheerorganisatie. Na afloop van zo'n project komt die behoefte vaak alsnog aan de oppervlakte, en dan is er wederom de vraag wie de tooling aanschaf, maar ook wie de regressietestset kan samenstellen.

## Toolkeuze

Over het algemeen ligt bij de selectie van een tool te veel nadruk op factoren die niet zo relevant zijn. Toevallige aanwezigheid van een tool of 'ervan gehoord hebben' zijn onvoldoende motivatie. Men moet eerst vaststellen wat er precies nodig is. Een inzicht in beschikbare testtools is daarbij onontbeerlijk. Bij soorten testtools kunnen we bijvoorbeeld denken aan tools voor:

- testuitvoering: herhalen van functionele tests;
- performance/stress testing: testen met grote aantallen;
- bevindingenbeheer;
- coverage;
- vergelijken;
- statische analyse: analyse van de code;
- testmanagement.

Er zijn nog veel meer types tools en ook veel verschillende leveranciers. Aan het houden van overzicht heb je bijna een dagtaak. Door het grote woud aan bijvoorbeeld opensourcetooltjes (zie [www.opensourcetesting.org](http://www.opensourcetesting.org)) stopt de zoektocht naar een bruikbare tool voor vele klanten en professionals al snel. Het is dan ook belangrijk om te weten wat u wilt bereiken met een tool. Wat is de businesscase voor een tool? Welke activiteiten wilt u automatiseren? Bedenk hier dat u taken die u wilt automatiseren ook met de hand zou kunnen uitvoeren. Als u goed over de businesscase nadenkt en weloverwogen keuzes maakt om tools aan te schaffen, dan is de implementatie van een tool al bijna geslaagd.

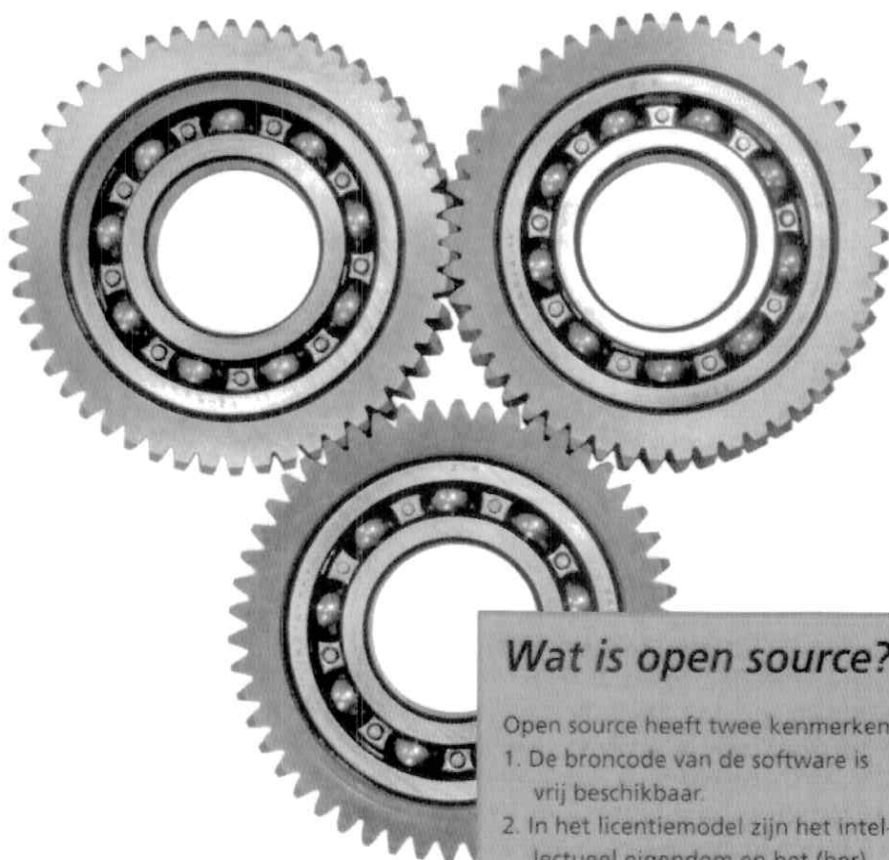
### Open source

In Nederland, en vooral binnen de overheid, is er een toenemende belangstelling voor het gebruik van opensourcesoftware (OSS) en open standaarden. Het is binnen de Nederlandse overheid zelfs verplicht eerst naar OSS te kijken alvorens closedsourcesoftware in beschouwing te nemen. Deze trend zien we in mindere mate ook in het ons omringende buitenland. Bij de Franse gendarmerie is men bijvoorbeeld het afgelopen jaar massaal overgegaan op Linux.

De hype van een voorkeur voor OSS heerst ook op het gebied van testtools. Die voorkeur is niet altijd een bewuste keuze uit een overtuiging, maar kan ook noodzaak zijn. Vooral bij overheden worden projecten gedwongen naar OSS te kijken. Als er geen budget aanwezig is voor het aanschaffen van commerciële

tools, zijn er weinig alternatieven. En als er wel budget is, dan steekt vaak de Europese aanbesteding de kop op, waardoor de aanschaf met zeker een halfjaar wordt vertraagd. Daar kan een project niet op wachten en dan moet men op zoek naar (tijdelijke) alternatieven om alsnog op een professionele manier het testwerk uit te kunnen voeren. Veel IT-consultants hebben hierdoor vaak OSS via projecten bij de klant geïntroduceerd. Niet altijd als vervanging van commerciële tools, maar in een aantal situaties ook als tijdelijk alternatief.

De bovengenoemde aanleidingen om naar OSS te kijken (verplichting, geen budget of lange tijdsduur) klinken niet ideaal, maar betekenen onzes inziens niet dat er geen goede testtooloplossingen binnen OSS beschikbaar zijn. Integendeel: het is eerder een manco dat de potentiële gebruiker zich nog niet voldoende bewust is van de mogelijkheden die er zijn op OSS-gebied.



### Wat is open source?

Open source heeft twee kenmerken:

1. De broncode van de software is vrij beschikbaar.
2. In het licentiemodel zijn het intellectueel eigendom en het (her)gebruik van de software en bijbehorende broncode dusdanig geregeld dat de licentienemer de broncode mag inzien, gebruiken, verbeteren, aanvullen en distribueren.

In dit artikel gebruiken we open source vooral in de context van 'gratis'. Closed source als antoniem van 'open source' gebruiken we als 'commercieel verkrijgbaar'. Er bestaan zowel bij open source als bij closed source verschillende licentievormen.

Het gemis aan kennis over de OSS-mogelijkheden leidt ertoe dat de optie van het gebruik van testtools vaak als "te duur" en daardoor als overbodig wordt beschouwd. Zelfs indien OSS wel als serieuze mogelijkheid wordt ervaren, is de eerste reactie nog afwijzend. De reden die daarvoor vaak wordt gegeven, is dat er geen gerenommeerd bedrijf achter zit en er dus geen zekerheid gegeven kan worden over de juiste werking en de

ondersteuning bij de tool. Men is er vaak "nog niet aan toe".

Duidelijk moet worden gesteld dat de commerciële tools vele malen volwassen zijn en daarmee vaak nog steeds de kwalitatief beste oplossing(en) bieden. OSS-testtools zijn vooral toepasbaar in de kleinere projecten of gedurende kortere perioden, waarin zij gemakkelijk, snel en doelgericht kunnen worden ingezet om een hogere mate van professionaliteit te krijgen.

De laatste jaren vindt er een lichte kentering plaats. Doordat bepaalde OSS-tools steeds professioneler worden, verdwijnen de 'mindere' tools en losse projectjes langzaam uit het zicht. Ook blijkt dat door het steeds weer hergebruiken van dezelfde OSS-testtools bij verschillende klanten de keuze eenvoudiger wordt. OSS-projecten worden ook steeds meer gesteund door commerciële bedrijven, wat ervoor zorgt dat de ontwikkelaars van deze tools daadwerkelijk betaald worden voor het verder ontwikkelen of verbeteren van de tools.

## Project

OSS-tools kunnen heel goed passen bij de ontwikkeling van een systeem (een project). Als er tools nodig zijn, is er bij een project namelijk de neiging om die tools te kiezen die de minste inspanning vergen om het beoogde resultaat te bereiken.

Dat hangt samen met de manier waarop een project wordt gestuurd. Die is weergegeven in figuur 1, naast de manier waarop in een beheersituatie wordt gestuurd. In beide situaties is er

een spanningsveld tussen tijd, kosten en kwaliteit. Projecten sturen over het algemeen op tijd en kosten; er is immers een projectplan opgesteld en daarin zijn aan deze schaarse factoren 'tijd' en 'geld' eisen gesteld (bijvoorbeeld een maximaal budget en/of een maximale doorlooptijd of minimaal het streven daarnaar). Kwaliteit is bij een project wel belangrijk, maar toch minder, zeker als er spanningen optreden en er keuzes gemaakt moeten worden als gevolg van tegenslagen.

Voor een groot project (in omvang en/of doorlooptijd) is closedsourcesoftware aan te raden. Er is dan in feite sprake van een beheersituatie, waar deze tools de voorkeur hebben.

## Beheer

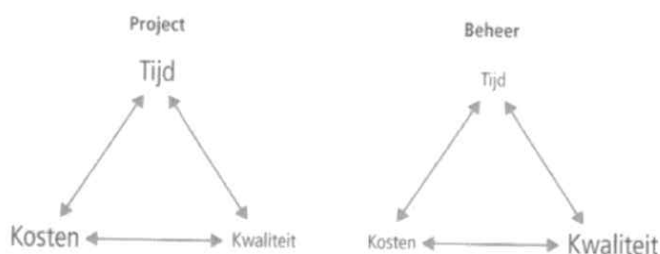
Beheer heeft als belangrijkste factor een goed product. Een hogere kwaliteit leidt tot een eenvoudiger implementatie, minder fouten in productie, minder herstel- en beheerkosten en meer tevreden gebruikers die vertrouwen in het geautomatiseerde systeem en de beheerorganisatie hebben. Beheer denkt op de langere termijn en houdt niet van verrassingen maar juist van stabiliteit. Met commerciële tools zijn verrassingen over het algemeen beter uit te schakelen; er is in ieder geval een aanspreekpunt, ondersteuning en continuïteit. Als beheerder zit je ook niet te wachten op projecten die allemaal andere tools gebruiken; standaardisatie vastgelegd

## Volwassenheid open source

Capgemini ontwikkelde het Open Source Maturity Model (OSMM), dat kijkt naar een aantal belangrijke kenmerken bij het meten hoe volwassen een opensourceproduct is.

- **Product**  
Hoe lang bestaat de software al? Zijn er al verschillende licentievormen? Is er één individu mee bezig of zit er een organisatie achter? Wie is er allemaal enthousiast over het product? Individuen? Community's of bedrijven? Is er een community? En hoe is die georganiseerd?
- **Integratie**  
In hoeverre kan er met scripts en plug-ins gewerkt worden? Hoe goed kan het product samenwerken met andere producten? In hoeverre werkt het product met algemene standaarden?
- **Gebruik**  
In hoeverre is de support geregeld? In hoeverre zijn er al dienstverleners die je kunnen helpen?
- **Acceptatie**  
Hoe uitgebreid hebben de gebruikers zich georganiseerd? Hoe groot is het marktaandeel?

Bron: OSMM, Capgemini, 2005



Figuur 1 Verschillen in sturing

in richtlijnen is gewenst. De extra investering voor de tool verdient je terug met verlaging van de operationele kosten. Een aanvulling met OSS voor specifieke zaken of voor een korte periode valt in bepaalde situaties te overwegen.

De verschillen tussen projecten en beheer zijn samengevat in tabel 1.

Nu kan beheer afwachten wat de projecten opleveren en dat accepteren, maar ze kunnen ook daadwerkelijk aan Accepteren doen. Bij Accepteren horen acceptatiecriteria en dat geeft beheer de mogelijkheid om eisen te stellen aan projecten.

Eisen stellen aan een project is mooi, maar dat betekent in feite ook eisen stellen aan de eigen beheerorganisatie. Kennis van tools of werkwijzen is dan een vereiste om te kunnen accepteren en onderdelen in beheer te kunnen nemen. Beheer is ook te outsourcen natuurlijk, maar beheersbaarheid niet!

#### De toekomst

Welke ontwikkelingen op het gebied van testtools staan ons de komende jaren te wachten?

#### Model-based testing

Al meer dan tien jaar staat de academische wereld klaar met het idee van model-based testing. Het concept en de tooling zijn inmiddels beschikbaar, maar

de wens om de werking in de praktijk te bewijzen is in de businesssoftwaremarkt nog beperkt. Veranderingen zijn voorzichtiger waarneembaar.

#### Zekerheid bij grote volumes

De wens om meer zekerheid over de werking van software te verkrijgen, leidt ertoe dat organisaties, zoals bijvoorbeeld de Belastingdienst, zekerheden willen inbouwen dat grote hoeveelheden data correct verwerkt worden. De enorme hoeveelheden zijn 'handmatig' niet te controleren. Tooling is dan een noodzaak.

#### Professionele testcenters

Testcenters met de tooling standaard in huis worden door bijna alle grote IT-bedrijven aangeboden. Idee is hier dat de klant behoefte heeft aan het uitbesteden van werk. Zelf kennis opbouwen over tools is en blijft lastig voor veel klantorganisaties. De drempel om tools te gebruiken is op deze manier impliciet en expliciet bij de leverancier gelegd. Zij zullen dat doen wat voor een project effectief en efficiënt is.

#### Allianties

Allianties tussen verschillende marktpartijen zijn steeds gangbaarder. De grote partijen zoeken elkaar op om vervolgens samen te opereren. Hierbij ontstaat een mooi spanningsveld tussen de onafhankelijkheid van de consultancybureaus

en de salesgedrevenheid van de test-toolleveranciers. Hierin heeft u als klant een grote verantwoordelijkheid, die u niet kunt afwentelen. De bureaus geven adviezen en u beslist. Zeker bij de overgang van project naar beheer.

#### Service-oriented architecture

Service-oriented architecture is een technologische ontwikkeling die ook het testen beïnvloedt. De wens om in uitwisselbare componenten te denken is niet nieuw, maar krijgt nu wel veel aandacht. Het testen moet aan deze architectuur aangepast worden met andere methoden, technieken en testtools. Bijvoorbeeld tools die ondersteunen bij het testen van services. Bedrijven die hier al jarenlang mee bezig zijn, zijn bijvoorbeeld Itko en Parasoft, die ook door Gartner als zodanig onderkend worden. Uiteraard zijn de grote softwareleveranciers als HP, IBM en Compuware niet achtergebleven.

#### Conclusie

Bij projecten hebben opensourcetools de voorkeur. Voor beheer daarentegen is closedsourcesoftware geschikter. Hierop zijn uiteraard uitzonderingen mogelijk. Testtools zijn nog niet uitontwikkeld. Zij zullen steeds meer gebruikt worden, maar het zal nog wel minstens tien jaar duren voordat het testproces volledig door geautomatiseerde hulpmiddelen ondersteund wordt.

Beheerorganisaties zullen een leidende rol moeten claimen. Het is daarbij van belang de markt te kennen en de ontwikkeling van testtools te volgen. Door eisen te stellen aan de op te leveren producten vanuit projecten, kan de beheerorganisatie daarna zelf ook de gewenste kwaliteit aan zijn klanten leveren. Een afwachtende houding leidt zelden vanzelfsprekend tot de optimale invulling.

Reinder Otter (reinder.otter@capgemini.com), Maurice Siteur (maurice.siteur@capgemini.com) en John van Veen (john.van.veen@capgemini.com) zijn werkzaam bij Capgemini.

Project	Beheer
veranderen/ontwikkelen	behouden/overnemen
snelheid	zekerheid
goedkoop	kwaliteit
besluitvorming	vertrouwen
business case	beheerbaarheid
korte termijn	lange termijn
voorkeur voor open source	voorkeur voor closed source

Tabel 1 Project vs. beheer