



thema

De laatste jaren is het verplaatsen van IT-activiteiten naar lagelonenlanden als India steeds populairder geworden. Naar verwachting zal deze trend zich de komende jaren voortzetten en versterken. Eén van de activiteiten die de komende jaren in toenemende mate offshore zal worden uitgevoerd, is testen. Veel organisaties die deze stap nemen, zijn echter onvoldoende voorbereid en hebben veel te hoge verwachtingen.

Offshore testen

Mooi van ver, maar verre van mooi

Testen als IT-activiteit wordt vaak onderschat, terwijl de voordelen van offshoring juist worden overschat. Daardoor zou bij velen de gedachte kunnen postvatten: "testen is hier een probleem, laat India er maar zorg voor dragen". Organisaties die deze stap zetten, stellen zich echter aan een aantal forse risico's bloot.

PROBLEMEN Op dit moment worden verschillende testmethoden al op vele manieren offshore uitgevoerd. De problemen die ondervonden worden, komen vaak

situatie waarbij er een taalbarrière bestaat, de afstand zeer groot is en cultuurverschillen kunnen zorgen voor misverstanden, vindt deze compensatie maar gedeeltelijk plaats.

Gebrek aan businesskennis

Het gebrek aan kennis van de business is een probleem dat verergert naarmate de afstand tot Nederland groter is. Goede en uitgebreide specificaties kunnen het ontbreken van specifieke businesskennis slechts gedeeltelijk opvangen. Sommige kennis wordt vaak als impliciet aanwezig verondersteld. Goede voorbeelden daarvan zijn het werk van notarissen en het verstrekken van hypotheek, maar ook iets schijnbaar triviaals als het formaat van een Nederlandse postcode. Offshore testers en programmeurs doen dan aannames die nogal eens onjuist blijken te zijn.

Het is noodzakelijk de verschillende onshore en offshore testsoorten goed op elkaar af te stemmen en te bewaken

ook voor bij een regulier testtraject, maar alleen al door de afstand (fysiek of in andere vormen) zijn deze zwaarder en een stuk lastiger te beheersen. Enkele voorbeelden zijn:

Garbage In - Garbage Out (GIGO)

Testen is vergelijken. De softwaretester vergelijkt de software (het testobject) met een referentiekader in de vorm van requirements, ontwerp of specificaties, de testbasis genaamd. Heel belangrijk hierbij is de kwaliteit van de testbasis. Wanneer deze van slechte kwaliteit is (garbage in) dan zal de output ook van slechte kwaliteit zijn (garbage out). Bij een testproject waar de verschillende partijen dicht bij elkaar zitten, dezelfde taal spreken en dezelfde achtergrond delen, kan een gebrek aan kwaliteit van deze input worden gecompenseerd door korte en snelle communicatielijnen. Bij een offshore

Cultuur

Al bij nearshore trajecten kunnen problemen ontstaan door cultuurverschillen. Zo vinden mensen in de mediterrane landen ons Nederlanders vaak bot en onbeleefd in het zakendoen, terwijl wij hen onduidelijk vinden. Ook in India manifesteert het cultuurverschil zich in de omgangsvormen. Zo vinden veel Indiërs het onbeleefd om direct 'nee' te zeggen. Ze zullen proberen om op een andere, een beetje omslachtige maar wel positieve manier te vertellen dat iets niet kan. Voor de onwetende toehoorder kan dat dan heel gemakkelijk als een 'ja' worden opgevat. Dit cultuurverschil kan zich ook manifesteren in de bevindingenregistratie, omdat het als onbeleefd wordt gezien om kritiek te hebben op andermans werk. Bepaalde bevindingen worden niet als bevinding gezien, maar meer als het onvermogen van de tester om te begrijpen wat de programmeur gebouwd heeft.

Ontwikkelen en testen bij één organisatie

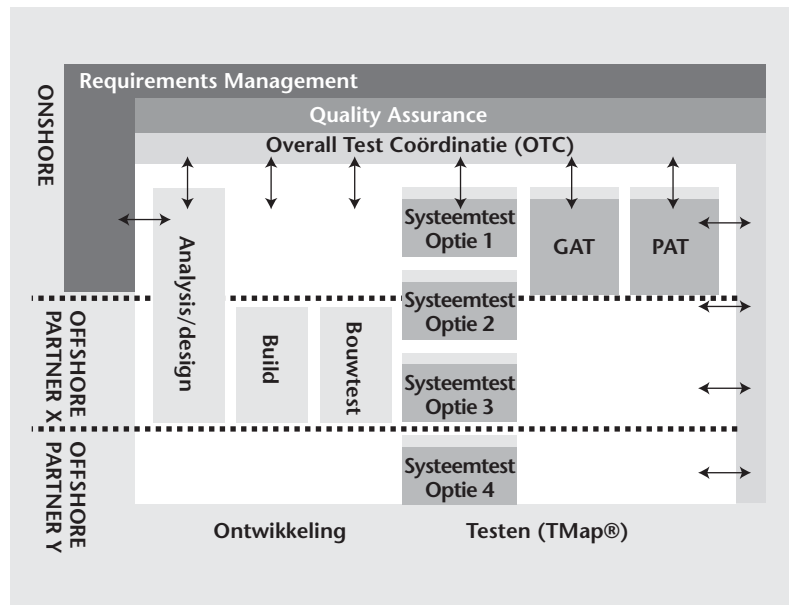
Veel organisaties zijn geneigd om het ontwikkelen van de activiteiten en het testen bij één en dezelfde offshore-organisatie te beleggen. Het grote gevaar dat hierin schuilt, is het 'vuile was'-probleem. Testers van hetzelfde bedrijf als de ontwikkelaars vinden het moeilijk om bevindingen te doen, want daarmee leveren ze tevens kritiek op hun collega's. Ze leggen die kritiek ook vast in rapportages, zodat de buitenwereld er zicht op heeft.

Testen als vak

Testen is een vak, maar wordt vaak nog gezien als iets dat je er even bij doet. Tegenwoordig is in Nederland het belang van testen en goede testers bij de meeste (grotere) IT-organisaties duidelijk. Bij offshore organisaties is dat nog lang niet altijd het geval en worden testteams gevuld met ontwikkelaars zonder enige training in, of ervaring met testen. Een dergelijk testteam is meestal niet in staat een gedegen testtraject op te zetten of duidelijk te maken wat ze nu hebben getest.

VISIE In onze visie volstaat het niet om de oplossingen enkel bij het testen te zoeken. Het is veel te duur en te tijdrovend om het GIGO-probleem in een test/herstel/hertest-cyclus op te lossen. Voor succesvolle offshoring moeten kwaliteits- en risicobeheersing breed ingevuld zijn. Daarvoor moeten de volgende vragen beantwoord zijn: *wat wil je bereiken, hoe ga je dat doen, doe je het op de goede manier en heb je je doel bereikt?* Het beantwoorden van deze vragen begint bij het vooraf goed definiëren van het doel (*requirements management*) en de beste manier waarop dit doel bereikt moet worden, namelijk een goed ingericht ontwikkelingsproces met uit te voeren activiteiten en op te leveren producten. Een goed geborgd proces, met de nodige controles op de daadwerkelijke toepassing ervan (*quality assurance*), levert hieraan een belangrijke bijdrage. Daarnaast moet de kwaliteit van de producten zo vroeg en zo goed mogelijk gecontroleerd worden: hoe eerder je een afwijking kunt opsporen, hoe beperkter de schade zal zijn. Een fout, die tijdens een eerste review van het functioneel ontwerp gevonden wordt, is goedkoper op te lossen, dan wanneer diezelfde fout tijdens het *testen* gevonden wordt. Om de verschillende testactiviteiten in goede banen te leiden, is coördinatie noodzakelijk (*overall test coördinatie*).

MODEL Voor het inrichten van bovenstaande beheersing van kwaliteit en risico's zijn een aantal activiteiten cruciaal in het proces van ontwikkelen en onderhouden van systemen. Door deze op gestructureerde wijze in te richten en uit te voeren kunnen de kwaliteit (en risico's) van offshore ontwikkeling en testen goed beheerd worden. De activiteiten kunnen weliswaar los van elkaar



FIGUUR 1. Een overzicht van de belangrijkste testactiviteiten.

worden uitgevoerd, maar gezamenlijk en gecoördineerd bieden zij integrale beheersing. Als de eerder genoemde maatregelen genomen worden, moet wel altijd gekeken worden naar de kosten ten opzichte van de baten, de verkregen zekerheid en de risicoreductie. Om in testtermen te blijven: 'no risk no test'. In figuur 1 zijn de belangrijkste activiteiten weergegeven.

Lage arbeidskosten kunnen een slecht testtraject nooit compenseren

We beperken ons in dit artikel tot de hiervoor meest in het oog springende zaken: Overall Test Coördinatie en de vier opties voor de Systeemtest. Requirements management, quality assurance, bouw- en acceptatietesten blijven verder buiten beschouwing (zonder dat dit enige afbreuk doet aan het belang van deze activiteiten!).

OVERALL TEST COÖRDINATIE De diverse testsoorten zoals programmatest, systeemtest en acceptatietest beslaan in de praktijk tussen de dertig en vijftig procent van het totale ontwikkelingsbudget. Wanneer één of meer van deze testsoorten offshore uitgevoerd wordt, ontstaan om eerdergenoemde redenen risico's. Feitelijk geeft de organisatie één van haar kwaliteits-'vangnetten' uit handen. Wanneer het offshore testen onvoldoende wordt uitgevoerd, dan worden in een laat stadium (acceptatietest of productie) een te groot aantal bevindingen gedaan, met projectuitloop of andere schade tot gevolg. Het is daarom absoluut noodzakelijk om de ver-

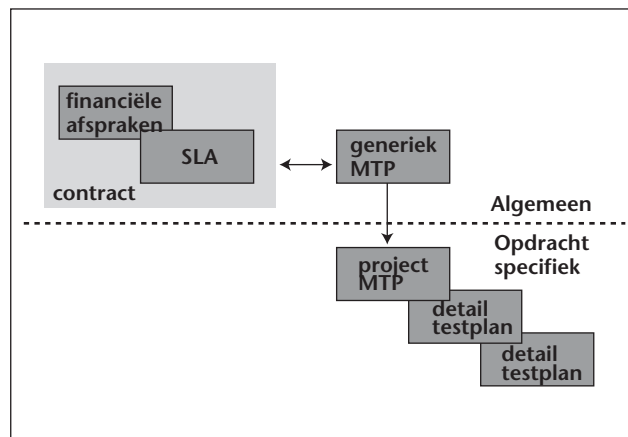
schillende onshore en offshore testsoorten goed op elkaar af te stemmen en te bewaken. Dit afstemmen, bewaken en bijsturen noemen wij Overall Test Coördinatie (OTC). Een hulpmiddel hiervoor is het opstellen van een mastertestplan dat alle testsoorten omvat. Standaard geldt een dergelijk mastertestplan voor één project. Offshoring overstijgt echter meestal een enkel project. De initiële investering begint meestal pas rendabel te worden bij opvolgende releases van het systeem, dus bij het onderhoud. Om te voorkomen dat elke release weer opnieuw afspraken gemaakt moeten worden over het wat en hoe van het testen, is daarom een extra mastertestplan in het leven geroepen: het Generieke Mastertestplan (GMTP). Hierin staan de algemene afspraken over bijvoorbeeld het testproces, procedures, communicatie, documentatie et cetera. Het GMTP is feitelijk een soort Service Level Agreement tussen opdrachtgever en opdrachtnemer. Per release of per project vindt dan een verdieping van het GMTP plaats in het zogenaamde Project Mastertestplan (PMTP), zoals het invullen van wat getest gaat worden, bemensing en mijlpalen.

Naast het inrichten van OTC zien we in de praktijk nog diverse andere maatregelen om de kwaliteit van het offshore testen te beheersen. Voorbeelden hiervan zijn het trainen en coachen van de offshore testers door ervaren testprofessionals, het monitoren van het testproces, het reviewen van testproducten en het hanteren van entry- en exit-criteria voor elke testsoorten. Een zeer handig hulpmiddel hierbij is het beschikken over een testmanagement-tool dat gedistribueerd gebruik ondersteunt (bijvoorbeeld web-based), vaak in combinatie met een bevindingenadministratie. OTC kan zo in één oogopslag zien welke testgevallen gemaakt zijn, welke uitgevoerd zijn en welke problemen geven.

SYSTEEMTEST In de systeemtest vindt met name de controle plaats of het gebouwde of aangepaste systeem voldoet aan de specificaties. Dit kunnen specificaties zijn in de vorm van een functioneel ontwerp, maar het kan ook een oude release op een ander platform zijn die als testbasis dienst doet. Omdat dit relatief recht-toe-recht-aan testwerk betreft (de specificaties zullen in het Engels opgesteld moeten worden om offshoring überhaupt zinvol te maken), is deze test een kandidaat om offshore te beleggen. Er zijn daarom op hoofdlijnen een aantal opties mogelijk:

- Optie 1: volledig onshore
- Optie 2: deels onshore, deels offshore
- Optie 3: offshore, bij dezelfde partij als de ontwikkelaar
- Optie 4: offshore, bij een andere partij als de ontwikkelaar

Binnen deze opties zijn er nog diverse mengvormen, bijvoorbeeld tussen wel of niet outsourcen. Zo kan



FIGUUR 2. In het Generieke Mastertestplan (GMTP) staan algemene afspraken over zaken als het testproces, communicatie en documentatie

ervoor gekozen worden om de test te outsourcen naar een onshore partner, die op haar beurt weer de testuitvoering offshore uitbesteed. In het hierna volgende zullen deze opties kort worden toegelicht.

• *Volledig onshore*

Dit is de meest conventionele aanpak en het meest geschikt wanneer men onzeker is over de kwaliteit van het ontwikkelde systeem en de risico's hoog zijn (bedrijfskritische systemen). Deze vorm biedt de meeste controle, maar is ook het duurst.

• *Deels onshore, deels offshore*

Een tussenvariant is om het plannen en specificeren van de test onshore te doen en de uitvoering offshore. Hiermee kunnen met de onshore aanwezige business- en testkennis goede tests worden gespecificeerd, die goedkoop offshore worden uitgevoerd. Daarbij is het essentieel om een webgebaseerd testmanagement-tool te gebruiken, dat zowel onshore als offshore te gebruiken is. Deze vorm biedt gemiddelde controle, en is met name geschikt voor het testen van systemen met een gemiddeld risico of wanneer de specificaties niet goed vastgelegd zijn.

• *Offshore, bij dezelfde partij als de ontwikkelaar*

Deze vorm komt in de praktijk vaak voor, maar biedt de minste controle mogelijkheden en is volledig gebaseerd op vertrouwen in de ontwikkelpartner. Om de eerder genoemde problemen onder controle te houden, moet veel energie gestoken worden in het beheersbaar houden van deze test. Hier kan het 'vuile was'-probleem optreden, waarbij de testers het bekend maken van hun bevindingen kunnen ervaren als het kritiek leveren op het eigen bedrijf. Vanwege de lage kosten is deze optie niettemin te overwegen wanneer het systeem niet bedrijfskritisch is.

• *Offshore, bij een andere partij als de ontwikkelaar*

Deze optie heft het probleem van kritiek leveren op het eigen bedrijf heel simpel op door het testen weliswaar offshore te beleggen, maar dan bij een andere partij. Bij outsourcing onshore gebeurt dit al heel

vaak. Het testteam kan in dit geval juist haar toegevoegde waarde aantonen door bevindingen te doen. Evenals bij de voorgaande optie moet zeker aandacht worden besteed aan het beheersbaar houden van de test. Er is het potentiële risico op inefficiëntie bij lage softwarekwaliteit: het nachtmerriescenario is een oneindige herstel-hertest-cyclus waarbij de software tussen beide partijen op en neer stuitert en de uren van beide partijen voor rekening van de opdrachtgever komen. Met goede afspraken is dit te ondervangen. Voor systemen met een gemiddeld risico is dit een efficiënte oplossing.

CONCLUSIE Betekent dit dat het offshore testen een slecht idee is? Nee, de lage lonen zorgen altijd voor een financieel voordeel. Maar tegelijkertijd is het belangrijk om te beseffen dat deze lage arbeidskosten een slecht testtraject nooit kunnen compenseren. Om offshoring een succes te maken, moeten onshore en offshore partijen bijzonder goed communiceren over wat en hoe te ontwikkelen en te testen. Feitelijk betekent dit een meer formele, beter gestuurde en ook andere manier van werken dan we nu vaak gewend zijn. Gelukkig hoeven we hiervoor geen

totaal nieuwe oplossingen te bedenken. De oplossingen bestaan, ze moeten alleen nog (beter) toegepast worden. Zo is transparantie van het testproces vereist. OTC, een GMTP en testmethoden als TMap bieden hiervoor de ver-

De diverse testsoorten beslaan tussen de dertig en vijftig procent van het totale ontwikkelbudget

eiste handvatten. De benodigde extra tijd en inspanning (door sommigen als overhead beschouwd) voor het transparant maken en houden van het testproces verdienen zich in de praktijk dubbel en dwars terug.

Dhanya-waadh! (Hindi voor: dankuwel)

Michiel Vroon en Tim Koomen zijn binnen Sogeti Nederland B.V. werkzaam bij Business Development voor de divisie Software Control, gespecialiseerd in testen en quality assurance.
