

TESTNET NIEUWS

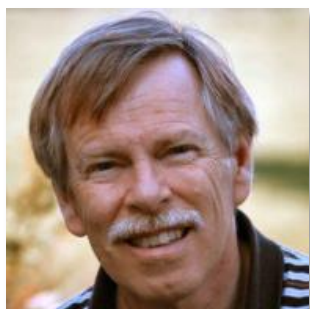
Mei 2012 • Jaargang 16 • Nummer 1

www.testnet.org secretaris@testnet.org



VAN DE REDACTIE

Door Hans van Loenhoud • tnn@testnet.org



'Eindelijk, daar is ie dan!' Met een zucht van verlichting presenteren we de nieuwe TestNetNieuws. We hadden grootste plannen met dit nummer. We zouden net als vorig jaar september een heel nummer aan het komende evenement

wijden, een glossy met uitgebreide aandacht voor het thema en de gepresenteerde onderwerpen. Daar hebben we veel effort ingestopt, zo veel zelfs dat ons reguliere maart-nummer daarvoor moest wijken. En toen ging de uitgever, Release, failliet. Dus helaas geen glossy op papier. Maar niet getreurd, wij testers laten ons niet zo gauw uit het veld slaan. De inhoudelijke bijdragen hebben we toch wel binnengekregen en ook als pdf ziet dat er heel aardig uit! Oordeel zelf, zou ik zeggen.

Ander heugelijk nieuws: naar aanleiding van een vacature in de TNN-redactie hebben we in de Nieuwsflits een oproep geplaatst voor nieuwe redacteurs. De respons was overweldigend, zowel in kwantiteit als in kwaliteit. We moesten zowaar kiezen en hebben de ploeg nu uitgebreid met drie kersverse redactieleden. In een volgende TNN stellen ze zich aan jullie voor. Ik reken erop dat we met deze versterkte club jullie lijfblad kunnen laten groeien en bloeien. Dus blijf inzenden ... ←

In dit nummer

Van de redactie	1
Van de voorzitter	2
15-jarig jubileum: een groots evenement	3
Tranaparantie = key!	4
In 15 jaar van ijsmeester naar weerman	7
Verbeteren van testen binnen agile/scrum	9
Weg met de testfase, of toch niet?	12
Testmethodieken; professionele bijziendheid	16
Praktische test review	21
Waarom en hoe als tester gebruik maken van NLP	25
Het ISACA RISK IT framework voor testers	27
Testen groeit mee in de wereld van ICT	32
Softwareapplicaties: goed gebouwd en toch niet af!?	35
iOS App testen	39
Testen van de API achter een mobile app	42
Van TPI naar TPI Next – 15 jaar	45
Testers staan boven de wet	48
Er zijn meerdere ijsbergen op de wereld	51
Opdrachtgeverschap en testen	55
Het is nu tijd om de blik naar buiten te richten	58
Testers, de moderne proefmeesters	61
A future of a tester: broaden or specialize	64
Combinatie functie requirements engineer en tester	66
Bestuursmededeling	68
Evenementen en Thema-avonden	69

TestNet Nieuws verschijnt eenmaal per kwartaal. Kopij aanleveren per e-mail aan de redactie. Het is niet toegestaan om de nieuwsbrief of delen eruit zonder bronvermelding over te nemen.

COLOFON

Redactie

Paul Beving
Hans van Loenhoud
Kees Blokland
Johan Vink
tnn@testnet.org

Bestuur

Michiel Vroon
John de Goey
Rik Marselis
Kees Blokland
Huib Schoots
Bart Watertor

Voorzitter

Penningmeester
Evenementen & Thema-avonden
Informatievoorziening & Beheer
Marktverkenning & Werkgroepen
Secretaris, 2e penningmeester

VAN DE VOORZITTER

Door Michiel Vroon • voorzitter@testnet.org



Vijftien jaar TestNet! Zo lang geleden is het initiatief genomen voor de oprichting van de vereniging TestNet. Martin Pol, Ingrid Ottenvanger, Erik van Veenendaal en Jos Trienekens zagen een groeiende belangstelling voor het testen van software in ons land. Daarom wilden ze platform creëren in de vorm van een vakvereniging voor en door testers. Op 9 juni 1997 was de oprichting een feit en werd de vereniging geboren.

Nu, vijftien jaar later, telt de vereniging rond de zestienhonderd enthousiaste leden. Testprofessionals die al jaren werkzaam zijn in het testvak en hun ideeën hebben over een juiste invulling van het vak, nu en in de toekomst. En dat is ook wat we als vereniging altijd goed in de gaten blijven houden: dat we niet alleen achteruit kijken, maar vooral vooruit!

Om dit te kunnen doen hebben we leden nodig, leden die op allerlei manieren bijdragen aan datgene waar TestNet voor staat. Leden die bijvoorbeeld meehelpen door het ontwikkelen van het vak in werkgroepen, leden die helpen met het organiseren van thema-avonden en evenementen en ook leden die helpen met het verzorgen van ons huisblad TestNetNieuws en vorm geven aan de website.

Maar natuurlijk ook, en die moeten zeker niet vergeten worden, de leden die invulling geven aan al deze activiteiten. Zij die artikelen schrijven en presentaties verzorgen, en zij die deze lezen dan wel bezoeken! Vanaf deze plek wil dan ook alle leden bedanken voor de mooie vijftien jaar die we hebben gehad, maar vooral ook die we nog gaan krijgen!

Bedankt! ←

15-JARIG JUBILEUM: EEN GROOTS EVENEMENT

Door TestNet Evenementencommissie • evenement@testnet.org

Op woensdag 30 mei 2012 vieren we met een groots jubileumevenement het 15-jarig bestaan van TestNet. Opgericht in 1997 is TestNet inmiddels uitgegroeid tot een vereniging met meer dan 1600 leden. Voor deze leden gebeurt van alles, vier keer per jaar een nieuwsbrief zoals degene die je nu leest. Een fraaie website met veel inhoud. En de bijeenkomsten. In het begin was er eens per kwartaal een bijeenkomst, inmiddels is het uitgegroeid tot negentien bijeenkomsten zoals thema-avonden, interactieve avonden, summer-school en het voorjaars- en najaarsevenement. Te veel om nog allemaal te bezoeken. Maar juist daardoor wel voor elk wat wils.



Voorjaarsevenement

Ook op het voorjaarsevenement is er weer voor iedereen wat interessants te vinden. Dat begint al met een keuze uit vijf boeiende tutorials/workshops in de ochtend. Daar leer je in drie uur van alles over een onderwerp dat jou interesseert. De middag begint met het vieren van het jubileum (taart!!) waarna de Jubileumboekcommissie (ook wel bekend als JuBoCo) het boek 'Bepaal je koers - toekomst en trends in testen' onthult en er op een boeiende wijze over vertelt. Daarna heb je keuze uit acht presentaties, over onderwerpen als de grootste misverstanden in de afgelopen vijftien jaar, trends in performance testen, testen in het onderwijs, het testen van apps en hoe je TMap Next nu echt moet toepassen. Het diner vindt plaats op de gezellige informatiemarkt waar de sponsors jou informeren over hun laatste nieuwtjes in ons vakgebied.

Keynote

De avond start met een keynote van Geoff Thompson (een bekende testexpert uit Engeland) en Bob van de Burgt (ex-voorzitter en erelid van TestNet) die ons een retrospectief geven en ons meenemen naar de 'times ahead'. Daarna kun je weer kiezen uit acht presentaties met uiteenlopende onderwerpen zoals het ISACA Risk IT Framework, Neuro Linguistisch Programmeren, Een Competente Center of Excellence, het testen van standaardpakketten en het verbeteren van de testgevallen. Aan het eind van de avond sluiten we uiteraard af met de beroemde testnetwerkborrel waar je onder het genot van een drankje en een bitterbal nog even bij kunt praten met je vakgenoten.

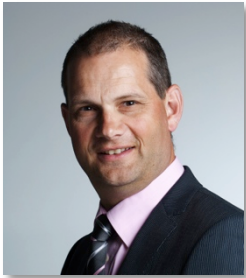
Vrienden

Nodig al je testvrienden uit om samen met jou naar het evenement te komen. Kun je er niet de hele dag bij zijn omdat je ook nog andere verplichtingen hebt, geen nood, voor leden is de toegang gratis, dus ook al kom je maar even langs, dan is het al zeer de moeite. En ken je iemand die nog geen lid is, maak haar of hem er dan op attent hoeveel 'value-for-money' het is als je voor 87,50 euro (2,50 korting bij automatische incasso) lid wordt van TestNet. Wil iemand echt geen lid worden, dan kan zij/hij voor 50 euro aan dit evenement deelnemen. Voorwaar geen geld (check maar eens wat je betaalt voor een dagje op een ander testevenement).

Samen maken we van dit jubileumevenement gelijk het grootste evenement ooit! ←

TRANSPARANTIE = KEY!

Door: Patrick Duisters • patrick.duisters@improveqs.nl

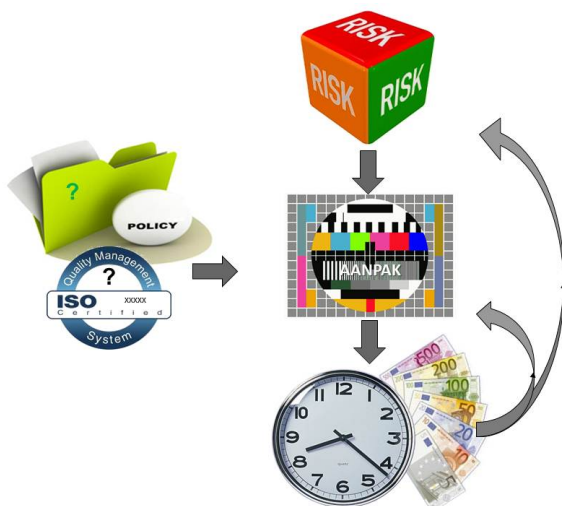


Wat mag worden verwacht van een gestructureerd en onderbouwd testproces dat de toets der kritiek kan doorstaan en in staat is om het management te ondersteunen bij de vrijgavebeslissing? Hier volgen de belangrijkste elementen van een volwassen testproces zonder direct formeel en bureaucratisch te worden! In de toekomst zullen toezichthoudende instanties meer en meer gaan meekijken in het toegepaste ontwikkel- en testproces. In medische en automotive industrieën zijn we daar inmiddels wel aan gewend. Maar ook in de technische en administratieve industrieën wordt toezicht steeds gebruikelijker.

Nu is het toezicht veelal nog gerelateerd aan veiligheid van kritische systemen, maar ook in outsourcing-situaties zijn er voorbeelden van projecten waar een instantie uit zich zelf of een externe partij gevraagd wordt om eens een kijkje in de (test)keuken te nemen (bijvoorbeeld door middel van een audit) en daar een oordeel over te vellen. Belangrijk is het om bij een dergelijke toets of audit om een aantoonbaar en inzichtelijk proces te hebben. Als je (te) veel moet uitleggen wat je hebt gedaan en waarom, dan wordt het als snel onduidelijk en onbetrouwbaar.

Alles vastleggen?

Nu zul je denken: lekker formeel en bureaucratisch zeker zo'n testproces! Alles vastleggen, veel documentatie, daar krijg ik geen tijd voor! Toch kun je ook met beperkter middelen een inzichtelijk testproces inrichten zonder direct formeel en bureaucratisch te zijn, maar wel zodanig dat een auditor én het management inzicht hebben en vertrouwen krijgen in een onderbouwd testproces. In mijn vijftien jaar ervaring met testen en audits van complexe en grote financiële, medische en andere technische systemen zoals medische DNA analyse en tunnels heb ik veel varianten van testprocessen gezien. In het algemeen zie ik veel typische testproducten gemaakt worden. Testplannen, testspecificaties en testrapporten zijn veelal beschikbaar. Testontwerpen ontbreken helaas vaak waardoor niet inzichtelijk wordt hoe de testgevallen in de testspecificaties tot stand gekomen zijn. Maar wellicht volstaat alleen het testontwerp al en kan een uitgebreide testspecificatie wel achterwege blijven. Mijn ervaring leert verder dat de verschillende testdocumenten regelmatig niet op elkaar zijn afgestemd. Ik vraag me dan wel eens af of ze echt 'informatie' verschaffen, of dat ze worden gemaakt omdat het moest of dat er templates zijn ingevuld.



De basis...

Het fundament van een aantoonbaar en minimaal testproces kan worden weergegeven in de figuur hiernaast.

Risico's

Het draait allemaal om risicobeheersing. Beperking van schade veroorzaakt door het onjuist functioneren van het (software) systeem. Financiële schade, imagoschade of erger nog: letsel. De eerste belangrijkste stap is identificatie en analyse van de risico's. Bijvoorbeeld door middel van een Product Risk Analyse (PRA). →

Niets nieuws natuurlijk, maar is er iets aantoonbaar vastgelegd, al is het maar één of twee A4'tjes? Zelf heb ik goede ervaringen met PRISMA®, maar heb je ooit nagedacht over Risk Poker? De tweede stap is het beperken van de risico's.

Testaanpak

Risico's kunnen vervolgens worden geminimaliseerd door middel van testen te laten zien dat die risico's zich niet voordoen. De test aanpak dient daarvoor afgestemd te zijn op de eerdergenoemde PRA. De risicomatrix zoals we die van PRISMA kennen geeft een duidelijk overzicht van de te testen onderdelen en hun risico-indeling. Samen met bijpassend gekozen testtechnieken en diepgang kan dit als aanpak al voldoende zijn. Is er sprake van een (test)beleid en/of een kwaliteitssysteem dat meer voorschrijft met betrekking tot testen, dan dient daarmee uiteraard rekening gehouden te worden in zowel de testaanpak, -specificatie en testuitvoer.

Specificatie en uitvoering

Uiteraard heeft de testaanpak invloed op de voorbereiding, specificatie en uitvoering in termen van tijd en geld. Dat begint al bij de keuze van de testtechnieken. Die keuze is soms moeilijk, maar vaak ook lastig door relatief weinig ervaring met toepassing van de testtechnieken. Een Quick Reference Card met een overzicht van de testtechnieken per risicokwadrant, en met de verschijningsvorm van de testtechniek kunnen daarbij al een eerste hulpmiddel zijn. Formele, grondiger en volledig gedocumenteerde testtechnieken kosten meer dan informele explorerende testen. Dat sluit echter niet uit dat de meer informele testen niet mogelijk zijn, ook in de veiligheidkritische omgevingen. Ook hier kan men prima bijvoorbeeld Exploratory Testing toepassen, mits de charters vooraf en in lijn met de testaanpak worden opgesteld en de uitvoering wordt gelogd, in welke vorm dan ook. Indien je weet hoe grondig en liefst ook met welke testtechnieken je wilt gaan testen, dan kun je ook betrouwbaarder schattingen maken. Gedetailleerde work break down structures zijn dan nauwkeuriger in te vullen, maar ook hier kan het informeler door bijvoorbeeld gebruik te maken van planning poker. Wel wat resultaten bewaren natuurlijk.

Evalueren en rapporteren

Verder is het belangrijk om regelmatig terug te kijken. Bijvoorbeeld door middel van feedback loops of meer Agile: retrospectives, ook in traditionele testtrajecten! Is de testuitvoering wel conform de afgesproken testaanpak en is de grondigheid in lijn gebracht met de risico's? En eveneens belangrijk: geven bevindingen tijdens testuitvoering reden om de PRA te herzien? Vinden we veel fouten in 'laag risico' gebieden en is het daarom verstandig om de testaanpak hierop bij te stellen. En last but not least: is wat we volgens de testrapportage hebben getest wel in overeenstemming met wat we in het testplan hadden bedacht? En zo niet, is daar een (vastgelegde) reden voor?

Volwassen?

In geval van outsourcing is het ook zeer wenselijk om een zekere aantoonbare volwassenheid te hebben. Denk daarbij aan TPI Next en TMMi als volwassenheidsmodellen. Het aardige is dat, indien je de eerder genoemde basis hebt geïmplementeerd, je een belangrijk deel van bijvoorbeeld TMMi Level 2 al aan het invullen bent. De strategie zit grotendeels ingesloten in de risk based aanpak. De planning en schattingen zijn afgeleid van de aanpak. Met de feedback loops worden stappen gezet met betrekking tot monitoring en control. En met testontwerp wordt een onderbouwde basis voor testuitvoer gelegd. Voor Level 2 ontbreekt dan in hoofdlijnen alleen nog de testomgeving als process area. Dat het proces past binnen methodes als ISTQB en TMap Next is bijna overbodig om noemen. →

Business

Uiteindelijk is het een businessbeslissing hoeveel men wil besteden aan inperking van de risico's, óf het accepteren van kosten ingeval van uitval. Wanneer men besluit een substantieel budget te besteden aan testen, dan wil men daarvoor op zijn minst ook inzicht in hebben. Een transparant test proces heeft dan sterk de voorkeur en enige vorm van aantoonbaar uitgevoerde stappen is dan zeer wenselijk. Zeker als er ook nog mogelijk een externe partij of een instantie komt bepalen of je met je systeem wel of niet de markt op mag!

Samenvattend

In de toekomst zullen toezichhoudende instanties meer en meer gaan meekijken in het toegepaste ontwikkel- en testproces. Een inzichtelijk en aantoonbaar testproces schept meer vertrouwen en hoeft niet per definitie formeel en bureaucratisch te zijn. Testontwerp verdient daarin meer aandacht en feedback loops zijn een belangrijk instrument om te kunnen bijsturen. En zonder heel veel 'waste' kun je toch al methodisch en volwassen bezig zijn! ◀

IN VIJFTIEN JAAR VAN IJSMEESTER NAAR WEERMAN

Door Rik Marselis • rik@marselis.eu  @rikmarselis



TestNet bestaat al weer vijftien jaar. TestNet is opgericht in het jaar dat voor het laatst de Elfstedentocht werd gereden. Sindsdien heeft ons vak een mooie ontwikkeling doorgemaakt. Toch zijn we er volgens mij nog niet helemaal, de communicatie naar de opdrachtgever kan hier en daar nog beter. Hoewel het alweer bijna zomer is, dacht ik hierbij even terug aan de parallel van de betrokkenen bij het testvak en hen die een rol speelden bij de deze winter net-niet-doorgegane Elfstedentocht.

In de besluitvorming rond het doorgaan van de tocht der tochten zijn verschillende mensen betrokken. Zo heb je de voorzitter van de vereniging De Friese Elf Steden die we kunnen aanduiden als de 'principal stakeholder'. Hij neemt het 'oan / net oan' besluit. Dat klinkt toch een stuk mooier dan go/nogo ;-)

Om zijn besluit te kunnen nemen heeft de voorzitter informatie nodig over het ijs. Vooraf is al bekend dat een ijsdikte van vijftien centimeter 'goed genoeg' is en dat de risico's dan 'acceptabel' zijn. De ijsdikte wordt gemeten door de ijsmeesters (dat zijn dus de testers). Boor een gaatje, houdt de meetlat erin en je hebt de waarde.

Van beperkte duur

Omdat in Friesland een vorstperiode meestal maar van beperkte duur is, kan de voorzitter niet rustig gaan wachten tot alle ijsmeesters melden dat het ijs dik genoeg is en dan de voorbereidingen voor de tocht opstarten. De voorzitter wil graag een dag of twee, drie van tevoren al besluiten dat de tocht doorgaat. Op dat moment roept een aantal ijsmeesters echter nog 'elf centimeter, dus de kwaliteit is niet genoeg en het risico dat mensen erdoorheen zakken te groot'. Daar is geen speld tussen te krijgen, maar voor de voorzitter is dit niet de volledige benodigde informatie.

Hij wil namelijk weten wanneer het ijs wel dik genoeg zal zijn. Daar heeft hij een andere specialist voor nodig, de weerman. Deze Piet (laten we hem voor het gemak even zo noemen ;-)) vertelt dat het koufront nog twee dagen stabiel blijft en dat er net als de afgelopen dagen meer dan tien graden vorst in de nacht is. De weerman heeft ook de gegevens van de ijsmeesters van de afgelopen dagen bewaard en vastgesteld dat het ijs de afgelopen nachten met minimaal twee centimeter per nacht is aangegroeid. Dus hij schat in dat dit ook de komende nachten weer zal gebeuren. Op basis van deze trendanalyse concludeert de voorzitter dat het ijs over twee dagen vijftien centimeter dik is. Daar hebben we wat aan, haal de schaatsen maar uit het vet!

In de afgelopen vijftien jaar zijn wij testers heel goede ijsmeesters geworden. Wij kunnen van steeds meer testobjecten op steeds meer kwaliteitsattributen steeds gedetailleerder aangeven wat de kwaliteit is en welke resterende productrisico's er zijn. Maar ik kom onder testmanagers nog te weinig weermannen tegen.

Goede weerman

Wat moet je als testmanager doen om een goede weerman te zijn? Het begint natuurlijk met het verzamelen van informatie over de gewenste situatie. Wat wil de opdrachtgever bereiken? Wanneer is de kwaliteit goed genoeg en welke risico's wil hij niet lopen? Welke informatie heeft de opdrachtgever nodig om zijn besluiten te nemen?

Vervolgens bepaal je welke gegevens er nodig zijn om op die aspecten te kunnen terugkoppelen. In de bovenstaande elfstedensituatie gaat het om ijsdikte, nachttemperatuur en ijsaangroei per nacht. →

In een testproject kun je denken aan het aantal uitgevoerde testgevallen in relatie tot het totaal aantal testgevallen, het percentage geslaagde versus gefaalde testgevallen, de requirements waarvan is aangetoond dat ze gehaald zijn en de risico's waarvan is aangetoond dat ze niet zullen optreden. Waarbij we de risico's uiteraard onderverdelen in ernstcategorieën, als er ergens een beetje water op het ijs komt is dat tenslotte veel minder erg dan wanneer een groep schaatsers onder het ijs verdwijnt.

Trendanalyse

Met gegevens alleen ben je er echter nog niet. Om de stap van ijsmeester naar weerman te maken, moet je periodiek een trendanalyse doen en daarover rapporteren. Met betrekking tot de requirements-afdekking kun je voorspellen wanneer alle must-haves getest zullen zijn. Aan de hand van de trend in het oplossen van bevindingen kun je voorspellen wanneer de belangrijke risico's zijn afgedekt.

Ook maakt een goede weerman gebruik van historische gegevens. Vanuit vergelijkbare projecten kan een testmanager informatie halen om voor het huidige project beter te onderbouwen wanneer het systeem goed genoeg zal zijn. Daarbij is bijvoorbeeld het gebruik van S-curves een mooi hulpmiddel. In organisaties waar voldoende historische gegevens zijn (en een redelijk stabiele werkwijze) kun je op het moment dat een kwart tot een derde van de testgevallen is uitgevoerd al een heel nauwkeurige schattingen doen van het totaal aantal bevindingen dat zal optreden en dus ook van de tijd die nodig zal zijn voor fixen en hertesten, wat weer leidt tot de voorspelling wanneer het systeem met voldoende zekerheid in gebruik genomen kan worden.

Tot slot nog een belangrijk voorbeeld dat wij testers moeten nemen aan de weerman: Wees niet al te bang om er eens een beetje naast te zitten. Negen van de tien keer voorspelt de weerman regen of zonneschijn correct, maar die ene keer dat je natregent terwijl je geen paraplu had meegenomen ben je boos op de weerman. Die zal dan wellicht zijn verontschuldigen aanbieden en doet de volgende keer nog beter zijn best. Bij zijn voorspellingen houdt ook de weerman natuurlijk rekening met risico's. Bij het voorspellen van de ijsaangroei zal hij liever wat aan de veilige kant blijven met zijn voorspelling, want door het ijs gezakte schaatsers heeft hij minder graag op zijn geweten dan natgerogende testers.

Als we allemaal de stap maken van ijsmeester naar weerman, kunnen wij ook in de komende jaren een gewaardeerde rol spelen in het speelveld van kwaliteitszorg en risicobeheersing in de IT. Zodat alle betrokkenen een goed gevoel hebben op het moment dat er geroepen wordt 'it giet oan'! ←

VERBETEREN VAN TESTEN BINNEN AGILE/SCRUM

Door Jeroen Mengerink • Jeroen.mengerink@polteq.com



In de vijftien jaar dat Testnet bestaat zijn het testvak en de context van testen flink veranderd. Naast het testen betekent dit uiteraard ook iets voor de bestaande testprocesverbetermodellen. Deze worden al jaren toegepast, maar sluiten ze aan bij Agile/Scrum? In de meeste modellen wordt uitgegaan van een klassieke (waterval) ontwikkel- en testmethode. De aandachtsgebieden die worden beoordeeld, sluiten aan bij de fasering van deze methoden. Juist de fasering gaat bij Agile/Scrum anders. Bovendien is niet alleen de fasering anders, maar ook de rol die testen speelt, is veranderd.

Een testmanager moet op een nieuwe manier kijken naar Agile trajecten en aan de testers in Scrumteams worden andere eisen gesteld dan in watervaltrajecten. Testen is nu een rol en geen functie! Dit vereist aanpassingen in de manier waarop de volwassenheid van het testen beoordeeld moet worden.

Volwassenheid

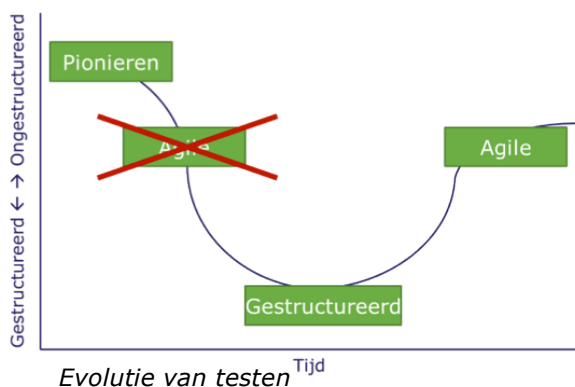
Om te kunnen verbeteren moet beoordeeld worden hoe volwassen een organisatie is op het gebied van Agile/Scrum én testen. Testen is immers een onderdeel van het geheel geworden. Hiervoor kunnen de volgende niveaus van volwassenheid worden gebruikt:

- Geen Agile/Scrum (men noemt het misschien wel Agile, maar in feite wordt er nog gewoon waterval gewerkt);
- Agile/Scrum proces is neergezet (alle voorgeschreven bijeenkomsten en artefacten zijn geïntroduceerd, er ontstaan kleine watervallen met steeds meer Agile kenmerken);
- Agile/Scrum wordt begrepen en toegepast (goed iteratief, incrementeel werk).

Voor een optimale verbetering moeten de niveaus worden bekeken vanuit een breder perspectief dan testen alleen. In de rest van dit artikel ligt de focus enkel op een aantal belangrijke testaspecten.

Testkennis

Eerste vereiste is natuurlijk een gedegen kennis van testen! De evolutie van het testen is in onderstaande figuur weergegeven. Eerst moest alles nog uitgevonden worden, daarna zijn de gestructureerde testaanpakken ontstaan. De volgende stap is het loslaten van de structuur om de verworven testkennis op een Agile manier te kunnen toepassen. Wanneer te vroeg begonnen wordt met Agile, zal de benodigde basis om goed te kunnen testen niet aanwezig zijn. Dit resulteert in het opnieuw moeten uitvinden van allerlei 'testwielen'.



De tester

Agile/Scrum stelt andere eisen aan testers dan watervaltrajecten. De tester moet veel meer taken op zich nemen dan hij gewend is. Planning en begroting werden altijd op een hoger niveau vastgelegd, maar moeten nu door de testers in het team worden meegenomen. Er is grote behoefte aan een proactieve instelling. →

Een testbasis is niet altijd direct voorhanden, dus moet de tester daar zelf achteraan gaan. De tester mag niet achter zijn bureau blijven wachten totdat de specificaties klaar zijn! In Agile/Scrum context moet de tester gesprekken met de andere teamleden initiëren en zorgen dat hij zo vroeg mogelijk kan beginnen met het maken van testgevallen. Niet zelden moet de tester zelf de testbasis creëren om deze vervolgens breder af te stemmen. De tester moet er ook aan bijdragen dat er per backlog item een duidelijke Definition of Done opgesteld wordt. Laat hier bijvoorbeeld enkele essentiële flows in opnemen met goed- en foutpaden. Dit helpt het hele team om te weten wat er nu daadwerkelijk opgeleverd moet worden en aan welke kwaliteitseisen dit moet voldoen.

Testmanagement

De testmanager moet in Agile/Scrum trajecten een faciliterende rol vervullen. Een rol die misschien iets minder testmanager en wat meer teammanager is. De Agile teams voorzien van mensen met testexpertise is natuurlijk niet onbelangrijk. Toch blijft op het vakgebied testen nog voldoende werk over voor een testmanager. De productbacklog wordt samengesteld door de business en een testmanager moet al op dat niveau gaan beginnen met het denken en communiceren over risico's. Op deze manier zullen afhankelijkheden tussen backlog items naar voren komen en kan beter bepaald worden wat er in de regressie en/of end-to-end test mee moet worden genomen. Op risicogebied houdt de testmanager dus het overzicht! Zeker wanneer er meerdere teams zijn is het belangrijk dat dit gebeurt. Een andere belangrijke taak, zeker bij grote organisaties, is het vastleggen van generieke testafspraken. Denk hierbij aan hoe om te gaan met kritieke bevindingen, eisen met betrekking tot testautomatisering en minimale, noodzakelijke rapportages. Hierbij kunnen twee niveaus van testafspraken gedefinieerd worden: organisatie breed en specifiek voor Agile projecten. Zo houdt de business meer inzicht in wat er gebeurt en blijft er een gevoel van controle. Puur Agile gezien lijkt dit de vrijheid van het team te beperken, maar praktisch gezien is dit zeker nodig.

Planning

Wat helaas vaak gezien wordt, is dat testen maar gedeeltelijk betrokken is bij het Agile werken. Het is in Agile misschien nog wel belangrijker dat testen vroeg wordt aangehaakt. Regelmatig worden planningsessies uitge-



Planningspoker

voerd zonder testers te betrekken of zonder testen een volwaardige stem te geven. Dit resulteert in een incorrecte inschatting van het testwerk en vaak ook het niet voldoende nadenken over de mogelijke risico's. Meest wordt planningspoker gebruikt om als team een inschatting te maken van de complexiteit/zwaarte van backlog items. Het is niet de bedoeling dat bij onenigheid gestemd gaat worden wat de waarde moet worden!

In alle belangrijke fasen binnen Agile/Scrum moet het team betrokken zijn. Daarnaast moeten bij voorbereidende werkzaamheden, zoals het samenstellen en prioriteren van de product backlog, de verschillende disciplines betrokken zijn. →



Testautomatisering

Een belangrijk element voor het testen binnen Agile/Scrum is natuurlijk testautomatisering. Om bij te blijven in het incrementele proces, waarbij de regressietest steeds meer zal omvatten, is testautomatisering noodzakelijk. Op dit gebied zijn vele verbetermogelijkheden. Goed automatiseren is niet zo gemakkelijk als het lijkt. De basis van goede testautomatisering zit in de architectuur (testframework) die daarvoor wordt opgezet en uiteindelijk in het bijhouden van de geautomatiseerde tests.

Business betrokkenheid

De organisatie als geheel moet de nieuwe manier van werken ook ondersteunen. Er is minder controle mogelijk en daar zal men (voornamelijk management) aan moeten wennen. De controle die altijd op basis van rapportages plaatsvond, moet nu in de samenwerking met het team gezocht worden. Hierdoor kan het team gestuurd worden in de richting die de business wil, maar heeft het team nog wel voldoende flexibiliteit. Agile/Scrum eist dus een hoge mate van betrokkenheid van de business. Een manier om de business te betrekken bij het ontwikkelproces is het leveren van heldere rapportage. Tools die de teams helpen bij het bijhouden van de voortgang en het vastleggen van defects bieden vaak goede rapportagemogelijkheden. Zorg ervoor dat deze rapportage dan ook ingericht wordt naar de wensen van de business.

Conclusie

Het is belangrijk dat men zich realiseert dat de focus en de aanpak van testen binnen Agile/Scrum anders ligt en dat daar rekening mee moet worden gehouden. Het team is verantwoordelijk voor kwaliteit en dus ook voor het testen. Dit ligt niet meer alleen bij de personen die toevallig een testfunctie hebben. Hoewel testen dus door meerdere personen gedaan wordt, wordt van de testers testexpertise verwacht en kunnen vooral zij bijdragen aan het verbeteren van het testen binnen Agile/Scrum. ←

WEG MET DIE TESTFASE, OF TOCH NIET?

Door Derk-Jan de Groot • Derk-JandeGroot@valori.nl



Er is veel discussie over hoe testen in de nabije toekomst georganiseerd gaat worden. Het testvak staat niet stil, vanaf het moment dat de ontwikkelaars gingen testen evalueert het vak. Toch lijken de ontwikkelingen de laatste jaren erg snel te gaan en voel ik onrust in de gelederen. Graag neem ik één van de discussie-threads op: de aparte testfase uitgevoerd door testspecialisten: is deze nog relevant?

De testfase is dood

Veel organisaties werken volgens de watervalmethode, waarbij er voor 'de final test' een uitgebreide testfase wordt ingeruimd. Daarnaast is een belangrijk kenmerk van 'gestructureerd testen' dat binnen een software-ontwikkeltraject meerdere perioden georganiseerd worden waarin een groep gespecialiseerde testers een bepaalde test uitvoert. De term testfase laat zich daarom goed definiëren als een groep van testactiviteiten die gezamenlijk worden uitgevoerd en aangestuurd. Voor de activiteiten binnen een testfase geldt bovendien dat deze hetzelfde doel nastreven, hetzelfde aandachtsgebied of systeemafbakening hebben. Vooraanstaande testers zoals James Whittaker en Gojko Adzic hebben in presentaties en artikelen herhaaldelijk aangegeven dat dergelijke testfasen niet meer levensvatbaar zijn. Een paar argumenten waarom 'De testfase dood is'.

Mosterd na de maaltijd

Er is weinig waarde in een kwaliteitsoordeel dat te laat komt. Binnen de watervalmethode is praktijk dat er helemaal aan het eind van het project een testfase wordt ingelast. Er wordt getest tot vlak voor de deadline, vaak loopt de werkdruk zo hoog op dat het testrapport postuum wordt geschreven. Het systeem is ondertussen al in productie genomen. Wat kan de organisatie in dat geval nog met opmerkingen en kanttekeningen? Wat moet het project met bugs die niet meer kunnen worden hersteld? Ook als er op eerdere momenten wordt getest, bijvoorbeeld de systeemtest, is er feitelijk sprake van 'controle achteraf', dus van mosterd na de maaltijd. De programmeurs hebben hun werk gedaan, willen starten met iets nieuws, maar moeten wachten totdat de testers een uitspraak gedaan hebben over de kwaliteit, gepaard gaand met een waslijst aan bugs.

Controle achteraf of zorgen voor betere kwaliteit?

De Customer experience is een aan terrein winnende KPI (Key Performance Indicator). Steeds vaker zetten organisaties de klantwaardering van het product centraal. Alhoewel bugs een bedreiging vormen voor de klantervaring devalueert de waarde van een vol bugtracking systeem snel. Het gaat niet meer het aantonen van de verschillen met de specificatie, het gaat om de tevreden klant. Agile ontwikkelingstechnieken richten zich dan ook veel meer op het gezamenlijk voorkomen en snel herstellen van fouten. De gebruiker wordt betrokken bij de ontwikkeling en coöperatie is belangrijker dan een formele specificatie. Dit reduceert de behoefte aan een onafhankelijk kwaliteitsoordeel achteraf.

De development cycle wordt korter

De levensduur van software wordt door steeds elkaar snel opvolgende innovaties korter. Logischerwijs dient daarom software ook sneller ontwikkeld te worden. →

Kent Beck geeft tijdens de USENIX Technical conference een duidelijke prognose. De deployment cycle zal de komende jaren afnemen van gemiddeld eens per kwartaal naar dagelijks of op uur basis. Voor testen heeft dit twee directe gevolgen. Ten eerste zullen testen snel uitgevoerd moeten worden. Je kunt immers niet een maand testen als de gehele release wekelijks wordt vrijgegeven. Ten tweede vervaagt de fasering, er wordt continu door iedereen getest. Er is dus geen ruimte meer voor een aparte testfase.

Een verschuiving naar operational assurance

We zien dat testen een activiteit wordt die door veel partijen wordt uitgevoerd. Ontwikkelaars binnen de sprint, Business Architecten bij het ontwerp en echte gebruikers tijdens het bètatesten. Kwaliteitsattributen als gebruikersvriendelijkheid, duurzaamheid en security worden steeds belangrijker en krijgen gedurende het gehele project aandacht. Een schril contrast met de traditionele testfase waarbij een groep onafhankelijke testers hun functionele testen uitvoeren met de adem van de deadline in hun nek.

Bovenstaande beschrijving laat een duidelijke verschuiving zien van formeel georganiseerde testfase(n), met name aan het eind van het ontwikkeltraject, naar een continu proces waarbij vele disciplines betrokken zijn. Linda Hayes geeft aan dat er een verschuiving is van quality assurance naar operational assurance, waarbij niet het kwaliteitsoordeel centraal staat, maar het ondersteunen van het operationele proces. Op basis van bovenstaande argumenten wordt duidelijk dat een van de 'slachtoffers' die deze verschuiving met zich meebrengt is de testfase is.

Argumenten voor een testfase aan het einde van het traject

De testfase is dus dood. Maar is dit wel zo? Er is namelijk ook een aantal argumenten aan te voeren die juist voor een aparte testfase achteraf pleiten.

Integratietesten

Hoewel het wenselijk is om zoveel mogelijk testen zo vroeg mogelijk uit te voeren, is dit niet altijd mogelijk. Unit- en systeemtesten controleren de kwaliteit tot een bepaald niveau. Door applicatie en toenemende koppelingen worden de systeemketens steeds langer. Door middel van goede simulaties en het werken met trusted components kan er veel getest worden, voordat systemen geïntegreerd worden, maar deze zullen de integratietest nooit helemaal vervangen.

De leverancier heeft andere belangen

Daar waar ontwikkelwerk uitbesteed wordt en mogelijk zelfs is geoutsourcet, ontstaat er een organisatorische grens. Aan weersijden van deze grens ontstaan vaak verschillende belangen. De accepterende partij heeft om redenen van politiek of geografische spreiding vaak geen werkelijk inzicht in de werkzaamheden van de leverancier. Dit maakt controle door de acceptant noodzakelijk. Bij voorkeur gedurende het traject en in samenwerking met de leverancier, maar acceptatie houdt toch vaak in dat er na oplevering nog even kritisch gekeken dient te worden.

Politics rule

Los van de acceptatievraag speelt politiek een rol bij het organiseren van testen. Steeds meer wordt het van organisaties verwacht dat ze voldoen aan compliance standaarden. Basel, SOx, SEPA om er maar een paar te noemen. Dit dwingt compliance-testen af en stelt eisen aan de formaliteit van de testactiviteiten. →

Daarnaast is het vaak wenselijk om gezamenlijke verantwoordelijkheid te dragen en te werken aan commitment. Dit kan bereikt worden door stakeholders en management bij het testen te betrekken. Een dergelijke testfase heeft een politieke doel.

Als het gewoon niet past

Zoals gezegd worden kwaliteitsattributen zoals security, performance, duurzaamheid en gebruikersvriendelijkheid steeds belangrijker. Ervaring leert dat deze specialistische testen vaak apart georganiseerd moeten worden. Performance testen, met name duurtesten, passen vaak niet binnen tweewekelijkse sprint. Als je Betatesten organiseert, levert een langere doorlooptijd een grotere dekking op. Redenen om deze testen apart te organiseren in een ... je raadt het al, aparte testfase.

Legacy

Bijna alle organisaties hebben te maken met legacy. Agile ontwikkelen, Continue integration en testen. Prima. Maar niet alle software leent zich voor deze wijze van ontwikkelen. Met name legacy systemen laten zich soms het beste op traditionele wijze ontwikkelen. Dit levert een aantal sequentiële stappen op en aparte testfasen. Volgens Ken Beck zijn er voor verschillende typen systemen verschillende testaanpakken nodig. Het kan dus effectief zijn om binnen een organisatie toch te kiezen voor verschillende testaanpakken. Daarnaast zijn er ook legacy organisaties. In deze organisaties bepaalt niet de techniek wat er mogelijk is, maar de cultuur en aanwezige kennis. Agile ontwikkelen vraagt de juiste expertise en mindset. Niet elke organisatie is daar aan toe.

Sommige systemen zijn levenskritisch

Bij levenskritische systemen gelden de hierboven genoemde argumenten nog sterker. Als er echt levens van afhangen zal de organisatie én zoveel mogelijk problemen vroegtijdig uitsluiten door testen te integreren in de ontwikkeling én objectieve testmomenten inbouwen. Hier gaan dus beide standpunten in elkaar op en bestaan ze naast elkaar.

Best of both worlds

We hebben gezien dat er argumenten zijn aan te dragen voor en tegen aparte testfasen in de softwareontwikkeling. Het gaat er niet om dat we als testers een keuze maken en krampachtig vasthouden aan de ons bekende testfasen. Het is ook niet nodig meteen radicaal de oude aanpakken helemaal in de ban te doen. Door de huidige ontwikkelingen verschuift er veel. Het is belangrijk voor testers om deze ontwikkelingen te volgen en na te gaan wat de consequenties zijn voor het testvak en de wijze waarop we ons werk doen. In mijn ogen wordt het vak daardoor kleurrijker, veelzijdiger en uitdagender. We krijgen er tools en opties bij. Als teststrategie betekent dit dat we goed moeten nadenken over welke bijdrage we leveren aan de organisatie en welke doelen we nastreven met onze activiteiten. Op basis daarvan kunnen we keuzes maken. Hierbij laten we oude en nieuwe denkbeelden samenkomen. Dit levert testers die in een organisatie meedraaien met ontwikkelaars om fouten te voorkomen en snel te detecteren. Dit levert ook testers die daar waar dat efficiënter is, bezig zijn met apart georganiseerde testfasen. Bijvoorbeeld in een fase waarbij we alle activiteiten rondom een risico groeperen. Bij een goede businessalignment levert testen informatie op die nauw aansluit bij de informatiebehoefte van de business. Onafhankelijk van wanneer de bepaalde testactiviteiten uitgevoerd worden kan het lonen om die activiteiten die bijdragen aan dezelfde inzichten en informatie apart te organiseren. De testcoördinator wordt zo een verantwoordelijke die namens de business zorg draagt voor inzicht en comfort op een of meerdere belangrijke aandachtsgebieden. →

De testfase is nog lang niet dood, maar hij zal steeds vaker anders gedefinieerd en georganiseerd worden. Ik vind dat prima, zolang we maar feeling houden met de behoefte van de organisatie, we ons continu scherp houden om zo efficiënt mogelijk maximale toegevoegde waarde te leveren en bijdragen aan operational excellence.

Referenties

[Hayes, 2010], Linda Hayes, 'You're Either On the Train or On the Tracks: Radical Realities Shaping Our Future', keynote op STARWEST 2010.

[Beck, 2010] Kent Beck, 'Software G Forces: The Effects of Acceleration'

[Whittaker,2011], James Whittaker, 'Pursuing Quality? You Won't Get There By Testing' (keynote EuroSTAR 2011).

[Adzic 2011], Gojko Adzic , 'Death To The Testing Phase' (keynote EuroSTAR 2011).

[JuBoCo,2012] Bepaal je koers, Toekomst en Trends in Testen, TestNet Jubileum Boek ←



TESTMETHODIEKEN; PROFESSIONELE BIJZIENDHEID

Door Leon Bosma • lbosma@kza.nl



De vijf grootste misverstanden

De eerste 'T' in TestNet staat voor 'Testen in de IT-wereld professionaliseren'. Professioneel werken bestaat uit systematisch en methodisch werken. Hoewel een methode een systematische aanpak vereist, brengt een vaste structuur risico's mee. Het grootste daarvan is dat er dogma's ontstaan en dogma's leiden vrijwel altijd tot misverstanden. Methodes voor testen zijn daarop geen uitzondering; er zijn vijf grote misverstanden ontstaan. Een veelgebruikte testmethode is TMap.

De vorm en inhoud zijn voor testers zo herkenbaar als het leesplankje voor iedere andere Nederlander. TMap is geschreven door de eerste kopstukken uit het testvak in Nederland. In de afgelopen jaren is het vervolgens meermalen aangepast op basis van opgedane en verzamelde ervaring. Zo heeft deze ervaring een vaste structuur gekregen. En die structuur heeft geleid tot misverstanden.



Testmanagement is noodzakelijk

'De primaire verantwoordelijke rol voor het opstellen van het testplan is de testmanager, soms ook testcoördinator genaamd.' (TMap). Vrijwel ieder project in softwarevoortbrenging heeft een testmanager om het testplan te maken. De basis van een testplan is de teststrategie. Voor een teststrategie worden keuzes

gemaakt over het doel van de tests. Toch is het maken van dergelijke keuzes niet zo ingewikkeld. Iedere oplossing in software steunt op drie pijlers: functionaliteit, geld en tijd. Om een project te kunnen realiseren moet tenminste een van deze pijlers worden vastgezet. De andere blijven dan flexibel.

Voor de teststrategie betekent dat de volgende zaken.

- Bij gefixeerde functionaliteit tijd en geld is flexibel: de beste teststrategie is dan om te testen op basis van de eisen en wensen voor de functionaliteit;
- Bij gefixeerde tijd, maar functionaliteit en geld zijn flexibel: de beste teststrategie is dan om te testen op basis van de risico's rond de oplossing;
- Bij gefixeerd geld, maar tijd en functionaliteit zijn flexibel: de beste teststrategie is dan om te testen op basis van gewenste opbrengsten van de oplossing.

Kortom, de keuzes zijn snel gemaakt en het vervolg is een handigheidje.

In alle gevallen stellen de belanghebbenden de prioriteiten:

- bij eisen en wensen volstaat een MoSCoW-lijst van alle eisen en wensen;
- bij risico's volstaat een overzicht van kans op en impact van ieder risico;
- bij opbrengsten volstaat een overzicht van kans op en omvang van iedere opbrengst. →

Vanaf daar kan iedere testspecialist prima bepalen welke technieken nodig zijn en zijn alle andere werkzaamheden van een testmanager exact hetzelfde als die van een projectmanager. Plannen maken, mensen selecteren, taken verdelen en voortgang volgen.

Kortom, de testmanager is als het pak van de keizer; het is niet echt. Dus laten we stoppen te doen alsof het bestaat.

Testtechnieken zijn afhankelijk van documentatie

'De fixatie van de specificaties is van groot belang. Ze vormen immers de basis voor zowel de testers als de ontwikkelaars...' (TMap)

Veel, zo niet de meeste, projecten in softwarevoortbrenging beginnen met het uitschrijven van het ontwerp. Dit ontwerp is daarna de basis voor alles wat er volgt, ook voor het testen. Methodische testtechnieken gebruiken het ontwerp, en aanvullende documenten, voor het voorspellen van het systeemgedrag. Toch blijkt er altijd een verschil tussen wat er beschreven staat en wat er daadwerkelijk wordt gebouwd. Testen op basis van documentatie richt zich op de overlap van het ontwerp en de feitelijke software. Hoe kleiner de overlap, hoe meer er door het voorgeschreven gebruik van de testtechnieken niet wordt gevangen.

Dan is er nog het verschil tussen wat er is bedoeld en wat er werkelijk in het ontwerp staat. De opdrachtgever heeft een beeld van de oplossing die hij wil. Dit beeld brengt hij vervolgens onder woorden, in zijn eigen woorden. Het ontwerp is de eerste vertaling van deze woorden.

Programmeurs zetten deze vertaling om in software. En ten slotte vertoont de software gedrag dat al of niet voldoet aan het eerste beeld. Ook dat verschil wordt niet gevangen door gebruik van testtechnieken op basis van documentatie alleen.

Testtechnieken zijn in feite gestructureerde vragenlijsten. Bij gebruikelijke inzet van deze technieken komen de antwoorden uit de documentatie. Toch kunnen testtechnieken uitstekend zonder documentatie. Het vraagt niet meer dan een verandering van bron. De tester stelt dezelfde vragen, maar haalt zijn antwoorden dan bij mensen. Bij voorkeur direct bij de opdrachtgever met het beeld van de gewenste oplossing. Hoe directer de bron van de antwoorden, hoe beperkter de interpretatie. Kortom, testtechnieken voor alleen de documentatie is als kijken met één oog; slechts een deel van het beeld en geen diepte.

Deep Thought en het juiste antwoord?

Het boek 'The Hitchhikers Guide to the Galaxy' (Douglas Adams) vertelt over Arthur Dent. Door een reeks van onwaarschijnlijke gebeurtenissen raakt hij op drift in het universum. Tijdens zijn reis ontmoet hij vele karakters en hoort hij vele verhalen. Eén van die verhalen gaat over de supercomputer genaamd 'Deep Thought'. Een groep pandimensionele wezens wil het antwoord op de ultieme vraag over het Leven, het Universum, en Alles. Zij willen weten wat de betekenis is van het leven. Ze construeren daarvoor een computer en noemen hem 'Deep Thought'. Hij moet het antwoord op de ultieme vraag over het Leven, het Universum, en Alles berekenen. 'Deep Thought' berekent gedurende een verloop van 7,5 miljoen jaar het antwoord: '42'. Maar niemand weet wat te doen met dit exacte antwoord, aangezien niemand weet wat de ultieme vraag nu eigenlijk is. Zonder de vraag kan niemand het antwoord begrijpen. Daarmee begint dus een nieuwe zoektocht; wat is de ultieme vraag over het Leven, het Universum, en Alles, waarop '42' blijkbaar het antwoord is?



Testen moet bevindingen opleveren

'Elke testontwerptechniek is gericht op het bereiken van een bepaalde dekking om bepaalde soorten fouten te vinden.' (TMap)

Testrapporten staat vaak vol met lijsten, staatjes en grafieken over bevindingen. De aantallen en aard van deze bevindingen bepalen dan of een softwareoplossing goed is. Dat wekt op zijn minst de suggestie dat goede software foutloze software is. Toch is foutloos maar zelden de kern van het beeld dat de opdrachtgever heeft. Software kan veel verschillende doelen hebben. Bepaalde functionaliteit, verminderde risico's, vergrote opbrengsten en vele andere. Ieder van deze doelen vergt een andere manier van validatie en verificatie. Een opdrachtgever heeft zorgen over het behalen van zijn doel en testen is in feite niets anders dan gestructureerd antwoorden zoeken op de zorg van de opdrachtgever. Niemand heeft iets aan telkens hetzelfde antwoord op verschillende vragen. Toch is dat wat bevindingenoverzichten zijn; telkens hetzelfde antwoord. Vergelijk het met het langverwachte antwoord van Deep Thought, zonder aansluiting tussen de vraag en het gegeven antwoord is het antwoord betekenisloos. Behalve de nietszeggenheid van het antwoord van bevindingen is er nog een ander risico. In China wilde de overheid de bevolking stimuleren om gevonden fossielen in te leveren. Daarvoor werd een beloning per fossiel uitgelooft. Het gevolg was dat grote fossielen eerst in kleine stukken werden gehakt en dan ingeleverd. Testen voor alleen bevindingen kan leiden tot hetzelfde proces. Het gaat dan al snel om de aantallen bevindingen en niet meer om de waarde van de vondst. De waarde van testen zit in de beslissingen die het ondersteunt. Zowel genomen als nog te nemen beslissingen. Testen meet het effect van die beslissingen, zodat zij degelijk kunnen worden onderbouwd. Kortom, testen moet inzicht geven. Inzicht in de effecten van de softwareoplossing op de functionaliteit, risico's en opbrengsten van het bedrijf.

Borat en het ongewenste effect?

(Telegraaf, 23-04-2012)

AMSTERDAM - Woedend waren de inwoners van Kazachstan toen de film Borat met Sacha Baron Cohen in 2006 verscheen. De Russische deelrepubliek draait nu 180 graden en bedankt de initiatiefnemers van de film. De minister van Buitenlandse Zaken laat weten dat het toerisme in het land is vertienvoudigd sinds de komedie in 2006 uitkwam. 'Ik ben dankbaar voor Borat, omdat hij heeft geholpen toeristen aan te trekken,' meldt hij aan persbureau DPA. De uitspraken van de minister zijn opmerkelijk omdat de Kazachse regering altijd grote bezwaren had tegen het personage Borat. Toen de film vijf jaar geleden uitkwam, waren de Kazach woedend over de manier waarop ze werden neergezet en werd de film zelfs verboden. 'Borat: Cultural Learnings of America for Make Benefit Glorious Nation of Kazakhstan' vertelt het verhaal van de Kazakse reporter Borat die de opdracht heeft gekregen een documentaire te draaien over de gebruiken in Amerika. De film schetst de Russische deelstaat als een achtergebleven land waar vrouwen weinig rechten hebben en homoseksuelen taboe zijn.



Testen kun je automatiseren

'Een tool dwingt tot een standaard manier van werken en hierdoor wordt de menselijke factor uitgeschakeld.'
(TMap)

Wanneer wordt gesproken over testtools gaat het al snel over geautomatiseerd testen. De voordelen lijken er te zijn; sneller en minder kans op menselijke fouten. Tegelijk ontstaat er een risicovol beeld over wat testen eigenlijk inhoudt. Softwarevoortbrenging is een proces van innovatie. Bij innovatie is vooraf niet precies duidelijk wat de uiteindelijke oplossing wordt. Vergelijk het met het ontwikkelen van een nieuw automodel. Er wordt getekend, gekleed, en geprobeerd. Er worden prototypes gebouwd en deze worden op allerlei manieren getest. Een prototype zal zich niet altijd gedragen zoals verwacht. Dat maakt het testen onvoorspelbaar. De tester past zich aan en zorgt dat hij toch komt met de nodige informatie. Automatisering gaat juist over herhaling en dus voorspelbaarheid. Handelingen worden geautomatiseerd om ze sneller en foutlozer te kunnen uitvoeren. Er is geen tool dat tijdens de uitvoering een analyse doet van onverwacht gedrag van het systeem. Er wordt door een tool niet geleerd, geen aandacht besteed aan mogelijke nieuwe situaties en als het gedrag maar iets afwijkt van het verwachte dan stopt de uitvoering. Automatisering heeft daarom feitelijk alleen maar zin wanneer het gedrag van het systeem al eerder is aangetoond en er dus geen onverwacht gedrag meer is. Bijvoorbeeld, wanneer het nieuwe automodel af is, start de productie. Aan de lopende band worden alle geproduceerde auto's op vele punten automatisch nagekeken om te bepalen of de parameters van iedere reproductie hetzelfde zijn als die van het prototype. Het gaat hier dus niet om testen, maar om controleren. Controleren of het systeem nog doet wat het daarvoor ook al deed (tijdens de test). Kortom, wat vaak geautomatiseerd testen wordt genoemd is slechts het geautomatiseerd opnieuw uitvoeren van testgevallen.

Testers worden goed door van elkaar te leren

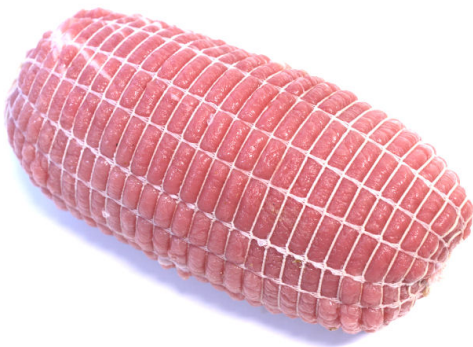
'Het goed kunnen uitvoeren van het gestructureerd testproces wordt door TMap ondersteund door een complete gereedschapskist.' (TMap)

Testmethodes en alle boeken die zij opleveren hebben één ding gemeenschappelijk; zij kijken terug. Hun doel is om vakgenoten een handvat te bieden voor de uitvoering van hun werk. Zij doen dat met de lessen uit het eigen vak, lessen die gisteren geleerd zijn. Drie grote bewegingen beïnvloeden op dit moment het werkveld van softwarevoortbrenging. Ten eerste is er de frictie tussen de wens sneller en flexibeler te leveren en de wens goedkoper te werken. Voor de vervulling van de eerste wens is er Agile, voor de tweede wordt gekeken naar standaardpakketten en uitbesteding. Ten tweede is er de versnippering van softwareoplossingen in steeds langere ketens bestaande uit steeds meer losse elementen. De elementen in huis zijn terug te vinden in Service Georiënteerde Architecturen en buitenshuis in een scala aan diensten in de Cloud. Ten derde is er de vervlechting van ICT en het primaire proces van bedrijven. In veel organisaties is er geen feitelijk verschil meer tussen beide; het primaire proces bestaat alleen nog voor het grootste deel uit ICT. Op het snijvlak van deze bewegingen doet een tester zijn werk. Om dat goed te kunnen doen, moet hij dus weten van aard, inhoud en richting van deze bewegingen. Dan volstaan de oplossingen van gisteren steeds minder. Dan volstaat alleen kennis van het eigen vakgebied niet meer. Het gaat niet meer over testtechnieken alleen. Best Practices vormen dan eerder een obstakel dan een hulp. Vooral als niet meer precies helder is wat de 'practice' tot een 'best practice' maakte. Kortom, testers worden goed in hun vak door over andere vakgebieden te leren en hun eigen 'waarheden' telkens ter discussie te stellen. ➔

Kapjes van de Rollade, best practice?

Een pasgetrouwd stel zit aan tafel. Zij kookt hun eerste gezamenlijke maaltijd. Trots dient zij een rollade op. Haar man kijkt in de pan en ziet dat de kapjes van de rollade zijn afgesneden. Hij vraagt aan zijn vrouw waarom ze dat deed? Haar antwoord is dat haar moeder dat ook altijd deed. Daar dacht ze verder niet bij na. De man is nieuwsgierig geworden en belt zijn schoonmoeder. Hij stelt haar dezelfde vraag. Het antwoord blijkt hetzelfde; omdat haar moeder dat altijd deed. Geïntregerd besluit de man om die zondag oma te bezoeken.

Na de eerste kop koffie snijdt hij het onderwerp van de rollade aan. Terwijl hij vertelt, wordt de glimlach van oma steeds groter. En dan komt haar antwoord. 'Wij hadden het vroeger niet breed. Zo had ik maar één braadpan waar ik erg zuinig op was. Het nadeel van die braadpan was dat hij net niet groot genoeg was voor een rollade. Dus sneed ik de kapjes van de rollade, zodat het toch paste.'



Kunnen de methodes dan beter overboord? Zeker niet, zij zijn een prima opleidingsmiddel. Een opleidingsmiddel zoals een leesplankje dat is. De reeks 'Aap, Noot, Mies' heeft velen leren lezen. Het opent een wereld van boeken en verhalen. Tegelijk leest niemand die boeken met het leesplankje op schoot. En niemand leest in die verhalen alleen de woorden van het leesplankje. Zo zou niemand moeten testen met TMap op schoot en zich vooral niet moeten laten beperken door de dogma's van een methode. Kortom, leer lezen, vul je woordenschat aan met alles dat je leest en vergeet nooit welke leesbril je draagt. ←

PRAKTISCHE TEST REVIEW

Door Jean-Paul Varwijk • varwijk@gmail.com  @arborosa



Heb je ooit het testen van iemand anders moeten overnemen? Of ben je lid van een Agile team, waar iedereen elkaars taken moet kunnen overnemen? Dan heb je vast wel eens meegemaakt dat je naar de testgevallen keek en dacht: 'Wat test ik hier nu eigenlijk?' Soms laat je dat geen andere keus dan zelf de testgevallen nog een keer uit te voeren of, erger nog, om ze zelf te gaan maken. En zelfs als dit niet nodig is, zou het een goed gebruik moeten zijn om elkaars testgevallen uit te voeren en te beoordelen. Voor testers bestaat het grootste gedeelte van ons vak uit het zoeken naar en het leveren van informatie.

Het gaat daarbij niet om zomaar informatie, maar om informatie met een doel. Het gaat bijvoorbeeld om informatie die ontwikkelaars defects laat oplossen, informatie die belanghebbende beslissingen laat nemen, informatie die iets zegt over ons werk en informatie die het mogelijk maakt om ook zonder onze aanwezigheid testgevallen te begrijpen, uit te voeren en te beoordelen. De bruikbaarheid van die informatie is in hoge mate afhankelijk van haar kwaliteit. Die kwaliteit zelf is niet alleen afhankelijk van de communicatieve vaardigheden of schrijfvaardigheden van de tester. Die kwaliteit is met name afhankelijk van de toegevoegde waarde en de bruikbaarheid van die informatie. Daarbij gaat het niet alleen om de inhoud, maar ook om de leesbaarheid, herkenbaarheid en begrijpelijkheid. Je hebt vast wel eens meegemaakt dat je de testgevallen van een andere tester moest uitvoeren. En daarbij heb je jezelf waarschijnlijk ook wel eens afgevraagd wat er nu eigenlijk getest werd, waarom er voor dit testgeval of deze testtechniek is gekozen of waarom er zoveel nodeloze of onbegrijpelijke stappen in het testscript zitten. Als oplossing voor met name deze problematiek heb ik het concept test review binnen het software testen bedacht en nader uitgewerkt.

Definitie: Test review is de systematische review (vaak peer review) van test gevallen of test resultaten. Test review is bedoeld om onduidelijkheden of fouten die gedurende testontwerp en testuitvoer worden gemaakt, te achterhalen en te verbeteren. Zodat zowel de kwaliteit van de testgevallen als de vaardigheden van de tester toenemen.

Context

Binnen deze definitie is een aantal kernbegrippen die ik binnen het kader van dit artikel aan bod wil laten komen. Het gaat daarbij om de volgende begrippen: systematisch; testontwerp; testuitvoering; onduidelijkheden; fouten; vaardigheden en kwaliteit. In het artikel maak ik veel gebruik van de aanduiding tester. Ik benadruk dat het mij daarbij om de rol van tester en niet zozeer de functie van tester gaat. Dat de professionele tester hieronder valt is evident, maar naar mijn mening valt eenieder binnen deze rol die zich in het kader van zijn werkzaamheden bezighoudt met testen. In de dagelijkse praktijk, en zeker in de Agile praktijk, zijn dat dus ook ontwikkelaars, business analisten, acceptanten of andere belanghebbenden die met enige kennis van zaken deelnemen aan het testproces. Test review is een systematische activiteit gebaseerd op het gebruik van een aantal methoden en technieken die in de praktijk hiervoor goed lijken te werken. Welke van deze methoden en technieken concreet in jouw situatie werken zul je zelf in de praktijk moeten bepalen. Dit hangt met name af van je ervaring met de betreffende methode of techniek, je individuele vaardigheden, de aard van de applicatie en de beschikbare tijd die je hebt.

→

Pair Testen

Pair Testen is een methode waarbij twee testers samen een vooraf bepaald testonderwerp of test(deel)object oppakken. Eén van de testers heeft daarbij om te beginnen de rol van de uitvoerder en de ander heeft de rol van de observator. De uitvoerder neemt plaats achter de pc en begint met het uitvoeren van haar tests en benoemt daarbij expliciet wat zij doet, waarom zij dit doet en zo mogelijk op basis waarvan. De observator beoordeelt gedurende de uitvoering voortdurend wat de uitvoerder aan het doen is, maakt daarvan aantekeningen in de vorm van opmerkingen en vragen, legt eventuele defects vast en stelt indien nodig vragen ter verduidelijking. Na een vooraf afgesproken periode, van meestal vijf tot tien minuten, wisselen beide testers van plaats en nemen elkaars rollen over. Na ongeveer een uur stoppen beide testers en beginnen aan een nabespreking waarbij de aantekeningen en



Pair testen (bron foto Wikipedia)

waarnemingen besproken worden en aan elkaar vragen kunnen worden gesteld ter verduidelijking. Indien nodig worden testsituaties die nadere uitleg behoeven nog een keer herhaald achter de pc.

Pair testen lijkt op het eerste gezicht sterk op een vorm van exploratory testen met zijn tweeën. In de praktijk zal dat vaak ook zo zijn omdat de exploratory testaanpak een goede basis is voor pair testen. Pair testen is echter ook in en kleinere opzet uitvoerbaar. Daarbij kan het scala aan keuzes van testtechnieken en bronnen die exploratory testen biedt, ingeruild worden voor een bescheidener selectie. Bijvoorbeeld door het te beperken tot maar enkele testontwerp-technieken zoals ad hoc testen of error guessing.

Walkthrough

Bij een walkthrough legt de schrijver van een bepaalde set testgevallen uit wat de inhoud van de testgevallen is. Hij geeft aan welk gebied van het testobject er mee afgedekt wordt en welke referenties hij heeft gebruikt. Hij licht toe welke technieken hij bij het maken van de testgevallen heeft toegepast en hoe deze, met welke data, moeten worden uitgevoerd. Het gaat daarbij niet alleen om een presentatie, maar om een interactieve sessie. Eén of meerdere deelnemers die als toehoorder of (liever) als actieve participant deelnemen aan de walkthrough geven



de schrijver feedback. Daarom is naast de schrijver en de deelnemer(s) nog een aantal rollen binnen een walkthrough. Zoals een notulist die tijdens de walkthrough eventuele wijzigingen, beslissingen en bevindingen vastlegt. Afhankelijk van het aantal deelnemers en hoe formeel de walkthrough wordt ingestoken kan de walkthrough worden aangevuld met een moderator voor het plannen en uitnodigen van de deelnemers en het voorzitten van de sessie. Of alle rollen noodzakelijk zijn en of deze door meerdere personen moeten worden vervuld, zal afhangen van de setting waarin de walkthrough wordt uitgevoerd. →

In het kader van een praktische test review zou ik zelf het aantal rollen zo klein mogelijk willen houden en daarbij de rol van bijvoorbeeld de notulist bij één van de deelnemers leggen. Het uiteindelijke doel van een walkthrough is om kennis te delen, testgevallen te toetsen op begrijpelijkheid, op uitvoerbaarheid en op vinden van eventuele fouten, omissies of uitbreidingsmogelijkheden.

Checklist

Een checklist kan worden gebruikt om te toetsen of de testgevallen, testuitvoering, testvastlegging of defectrapportage voldoen aan de daarvoor vooraf vastgestelde eisen. De eisen zelf kunnen zijn gebaseerd op specifieke projectcriteria, op regels binnen het team, op afdelingsregels, op bedrijfsregels of op van toepassing zijnde externe standaarden. Het kenmerk van een dergelijke lijst is dat het aantal elementen bevat die ervoor zorgen dat de testdocumenten begrijpelijk zijn, voldoen aan eventuele interne- of externe regels voldoende informatie bevatten om de in de documentatie vastgelegde informatie te kunnen duiden.

Over de schouder

Over de schouder is een wat meer informele techniek waarbij iemand bezig is met het ontwerpen of uitvoeren van testgevallen waarbij er zonder vooropgesteld doel of afspraak een tweede tester mee komt kijken. Het initiatief daarvoor kan bij beide partijen liggen en heeft als doel elkaar te helpen bij het oplossen van eventuele problemen, het beantwoorden of stellen van vragen of om elkaar, al dan niet ongevraagd, tips te geven of nieuwe vaardigheden te leren.

Voordelen van test review

Het uitvoeren van test review heeft een aantal grote voordelen. Enerzijds liggen deze voordelen in een verbetering van het testproces als zodanig en anderzijds in het vergroten en beter benutten van de aanwezige testkennis bij de testers.

Testproces

Test review biedt de mogelijkheid om bepaalde gebieden van de functionaliteit stapsgewijs af te dekken. Dit kan door het oppakken individuele features, op basis van scenario's, op basis van user stories of op basis van testcharacterters. Hierbij worden deze individuele delen van het testobject per sessie opgepakt en behandeld.

Naast dit aan de inhoud gerelateerd aspect heeft test review ook invloed op het gedrag van de tester. De wetenschap dat van de testgevallen zowel het ontwerp als de uitvoer door iemand anders beoordeeld gaan worden zal een tester stimuleren om daar meer aandacht aan te besteden. Er zal met meer discipline gewerkt worden aan het tijdig af hebben van de testgevallen, het volgen van eventuele regels en het toepassen van testontwerptechnieken. Tot slot is er nog een tijdsaspect. Door een beter management van tijd tijdens het ontwerp en de uitvoer van testgevallen kunnen deze, in doorlooptijd gemeten, sneller tot stand komen. De testgevallen kunnen, inclusief bevindingen, door de verbeterde begrijpelijkheid sneller worden uitgevoerd en afgerond. Ook eventuele ontbrekende of incomplete testgevallen zullen eerder worden ontdekt.

Testkennis

Doordat twee meer zien dan één en er met zijn tweeën (of meer) ook meer testkennis beschikbaar is, zal dit de kwaliteit van de testgevallen ten goede komen. →

Deze kwaliteitsverbetering komt tot stand omdat er sprake is van kennisoverdracht in het toepassen van testtechnieken, in domeinkennis, in vaardigheden met tooling en in het ontwikkelen van testideeën. Dit zal enerzijds gebeuren in de vorm van reviews tijdens het ontwerp, feedback op het eigen functioneren en het beoordelen van test vaardigheden. Anderzijds gebeurt dit door het benutten van elkaars creativiteit en het leren van elkaars fouten of het ontwikkelen van testvaardigheden voor zover deze bij de ander nog niet (voldoende) aanwezig zijn.

Nadelen van test review

Test review kent jammer genoeg niet alleen voordelen. De nadelen van test review liggen met name op het persoonlijke vlak van de tester en voor een kleiner deel in het proces.

De tester als mens

Ook een tester is maar een mens en als zodanig beïnvloedbaar door zijn omgeving. Door het gegeven dat er een test review uitgevoerd gaat worden kunnen testers op velerlei manieren beïnvloed worden of juist anderen beïnvloeden. Testers die gevoelig zijn voor druk van buitenaf kunnen in de context van een test review onderpresteren en minder productief zijn of een slechtere kwaliteit dan normaal af leveren. Ook kan een test review zogenaamd 'gewenst gedrag' oproepen. Hierdoor is hetgeen de tester tijdens de review laat zien niet representatief voor zijn normale werk. Hetzelfde geldt overigens ook voor de reactie van sommige testers op degene die de review gaat uitvoeren. Voor beide partijen is het daarom zaak om de 'ego's' buiten de test review te houden.

Testkennis

Hoewel hiervoor als voordeel genoemd kan een verschil in kennisniveau een test review ook nadelig beïnvloeden. Als het verschil namelijk te groot is, is de kans groot dat het verschil tijdens de review niet eenvoudig is te overbruggen en daarmee kan dit tot irritatie en frustratie leiden. Een voorbeeld hierbij is de toepassing van testontwerptechnieken. Lang niet elke tester zal een volledig repertoire van testontwerptechnieken beheersen of deze in de praktijk kunnen herkennen en beoordelen. Als voorbeeld geef ik het volgende lijstje van afkortingen met de uitdaging om bijbehorende techniek eens aan een mede tester uit te leggen; *GCT; SEM; DCT; SYN; EVT; EG; UCT; BTT; PCT; DGT; RLT; CKL*.

Testproces

Hoewel er voldoende mogelijkheden zijn om voordeel te halen uit test review, kost test review tijd. Bij bijvoorbeeld Pair Testen zal weliswaar de doorlooptijd korter kunnen zijn, maar het aantal manuren neemt hierdoor wel toe. De afweging van deze kostencomponent zal dus voorafgaand en tijdens de test review mee moeten worden gewogen.

Conclusie

Test review is volgens mij een zinvolle aanvulling aan het test repertoire. Het kan actief bijdragen aan het verbeteren van de algemene kwaliteit van de testgevallen, de testuitvoering, de testvastlegging en het afhandelen van defects. Het grootste voordeel ligt echter naar mijn mening niet zozeer in het verbeteren van de testdocumentatie. Het grootste voordeel is de toename van de samenwerking tussen de testers die tot gevolg heeft dat het kennisniveau van de testers kan stijgen, dat de testers elkaar beter begrijpen en daarmee niet alleen op dat moment, maar ook in de toekomst samen met het de overige leden van het (project) team een beter product kunnen afleveren. ←

WAAROM EN HOE KAN JE ALS TESTER GEBRUIKMAKEN VAN NLP?

Door Nicole Hilkemeijer • n.hilkemeijer@eclipseIT.nl en Rob Janssen • robjanssenprive@gmail.com



Neuro Linguïstisch Programmeren

Een goede tester kenmerkt zich mede door goede communicatie. Een tester is een spin in het web van programmeurs, analisten, projectleiders en eindgebruikers. Voor de tester van de toekomst geldt dat nog meer, omdat werken in Scrum en Agile teams steeds meer de standaard wordt. De levensvatbaarheid van deze methodes is in één woord communicatie.

Neuro Linguïstisch Programmeren (NLP) werkt toekomstgericht en leert je om je goed te voelen, te ontspannen en plezier te maken. Vanuit je eigen kracht zal je open en eerlijk andere mensen tegemoet treden. Met krachtig taakgebruik leer je hoe je inefficiënte stemmingen van jezelf en anderen kunt veranderen in vermogende stemmingen, waardoor je meer invloed hebt in je team waardoor je de teamprestatie als geheel kunt verbeteren.

Waar gaat NLP over?

NLP gaat in de eerste plaats over het goed leren gebruiken van je eigen brein. Over begrijpen hoe je brein in elkaar zit, hoe betekenissen door je brein gevormd worden en hoe je met die combinatie effectiever kunt acteren. Hierdoor heb je meer grip op je emoties, ontdek en verwezenlijk je je eigen doelen en ben je zelf degene bent die achter het stuur. Dit werkt twee kanten op. Aan de ene kant ga je met NLP beter om met tegenslag waardoor je minder last hebt van stress. Aan de andere kant leer je met NLP hoe je je eigen goede gevoelens kunt inzetten om mensen te motiveren en te inspireren. Je goed voelen blijkt daar een belangrijke component van te zijn.

Op de tweede plaats leert NLP je om beter te communiceren. NLP omvat twee communicatiemodellen, het Meta Model en het Milton Model, waarmee je enerzijds leert om zuiverdere informatie te delen en beter te luisteren naar wat de ander voor relevante zaken te melden heeft. Anderzijds leren de modellen je hoe je met hypnotische taalpatronen andere mensen kan beïnvloeden. Bij NLP gaan we natuurlijk uit van positieve beïnvloeding, hetgeen in een beroepsmatige (project)omgeving meestal uitermate handig is om toe te passen.

Meta model

Het Meta model kan een tester goed gebruiken bij het uitvoeren van statische testen, zoals het reviewen van requirements of een functioneel ontwerp. Hiermee kunnen de documenten beter beoordeeld worden om zo mogelijke bevindingen als gevolg van misinterpretaties te voorkomen. Daarnaast kan het model goed gebruikt worden om onwaarheden van anderen te herkennen en te kunnen weerleggen, zoals de welbekende uitspraak 'We moeten naar productie!'. Dit is volgens het Meta model een verloren maatgever, want waarom *moeten* we naar productie? Van wie moet dit? Waarom kunnen we niet op een ander moment naar productie? Is nu echt het enige mogelijke moment om naar productie te gaan?

Milton model

Het Milton model kan een tester goed toepassen om enerzijds te signaleren dat iemand hem of haar probeert te overtuigen van zijn/haar mening. Hier is op zich niks mis mee, zolang de argumenten inhoudelijk juist blijven. →

Echter, vaak genoeg wordt het Milton model misbruikt, al dan niet met opzet, om inhoudelijk slechte argumenten gemaskeerd over te brengen. Anderzijds kan een tester dit model toepassen om zo beter invloed uit te oefenen door mensen te overtuigen van datgene wat - voor jou als tester en voor de organisatie waarin jullie als team acteren - zo belangrijk is.

Het is belangrijk om te weten dat je slechter communiceert als je je slecht voelt, hoe goed je ook bent op het vlak van excellent communiceren op basis van NLP. Bij NLP is er sprake van twee fasen die elkaar opvolgen: eerst leer je je goed voelen, daarna leer je goed communiceren. Dat is voor elke tester natuurlijk een must-have! ←

HET ISACA RISK IT FRAMEWORK VOOR TESTERS

Door Jaap van der Leer • JaapvanderLeer@valori.nl



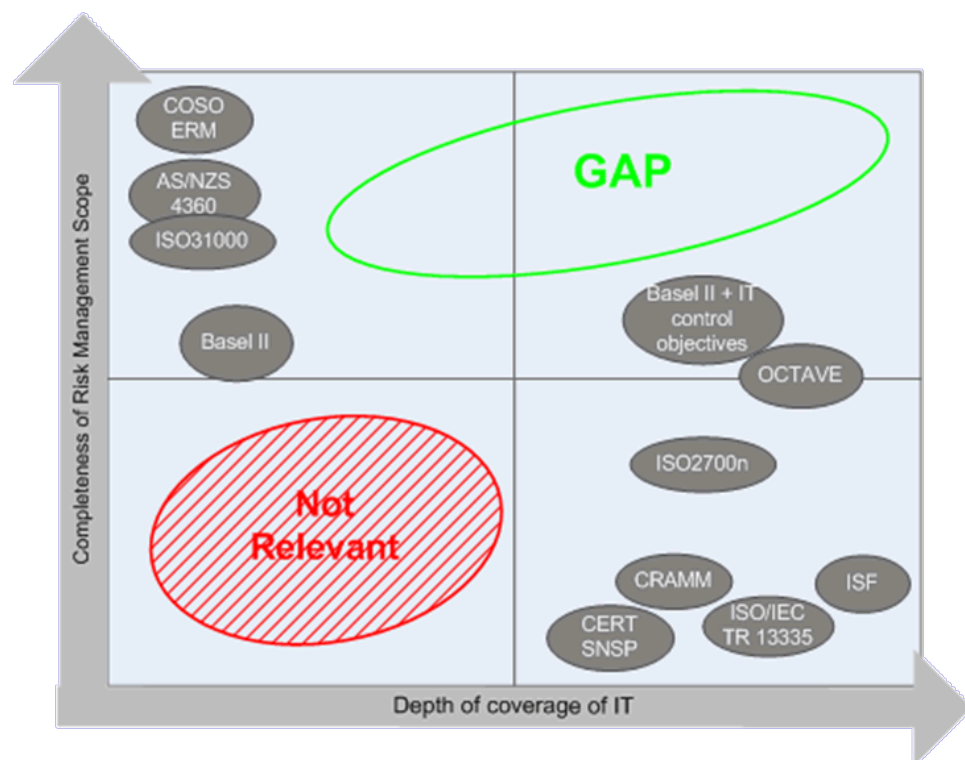
Testen baseren op risico's is al jaren gemeengoed. Nu zijn er risico's en risico's. Recent heeft ISACA het framework voor IT-gericht Risicomanagement uitgebracht, RISK IT. Met RISK IT beoogt ISACA risicodenken in IT-land op een hoger peil te brengen. Wij, testers, hebben natuurlijk ervaring met risico's meten, mitigeren, voorkomen, dus ligt het voor de hand eens goed naar RISK IT te kijken.

Met ideeën uit RISK IT van ISACA breng je Risico Gebaseerd Testen op een hoger plan, lever je meer waarde aan de stakeholders én kun je beter uitleggen wat je waarom doet.

Wat is ISACA?

In de wereld van testen is ISACA niet zo bekend. Toch is ISACA met zesennegentigduizend leden veruit de grootste vakvereniging ter wereld voor beheersing van IT. Wie kent bijvoorbeeld het CobiT framework niet? In CobiT worden de IT-processen generiek beschreven, een uitstekende basis voor IT-afdelingen om hun organisatie in te richten. In april 2012 jaar bracht ISACA CobiT 5 uit, waarin de IT-processen uit CobiT 4.2 worden verbonden met Business processen. Conform de doelstelling van ISACA *Vertrouwen in en waarde uit informatiesystemen* (zie <https://www.isaca.org>).

CobiT maakt op zijn beurt weer gebruik van andere frameworks, zoals dat voor Risicomanagement in IT, RISK IT. Anders dan bijvoorbeeld BASEL II en Solvency beschouwt RISK IT de processen in IT als gelijkwaardig aan Business processen. En anders dan eerdere op IT-risico's gerichte frameworks is RISK IT niet beperkt tot security en/of continuïteit. Er komen risico's bij, zoals dat IT niet de waarde levert die verwacht wordt, geen bijdrage levert aan bedrijfsdoelstellingen, of invloed heeft op bedrijfsrisico's als kredietrisico, marktrisico's of operationele risico's.



ISACA acht dit alles zo belangrijk dat ze voor risicomanagement een aparte certificatie in het leven riepen, CRISC.

→

Principes van RISK IT

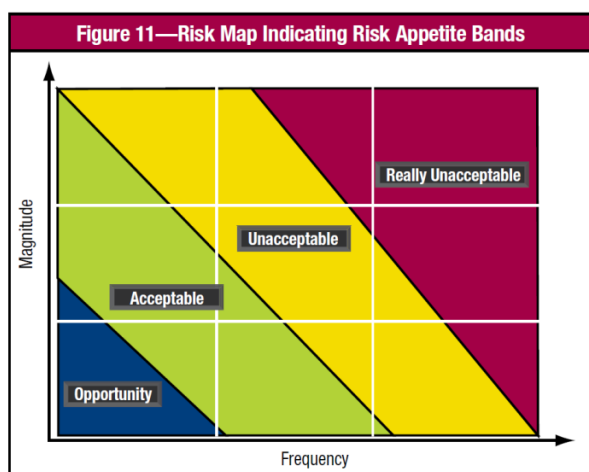
Als uitgangspunt, en dat is innovatief, onderkent RISK IT dat een risico tevens als kans kan worden beschouwd. Het gaat niet puur om mogelijke verliezen te beperken, het gaat óók om winsten binnen te halen. Daarmee krijgt IT een andere uitstraling, van kostenpost tot generator van omzet.

Een CRISC-gecertificeerde risicomanager is niet alleen uit op bevechten van storingen, inbraken en uitloop van projecten. Zijn of haar aandacht gaat in toenemende mate uit naar inzet van IT voor innovatie, marktaandeel, communicatie, en een zo hoog mogelijke opbrengst daarvan. Daardoor is het contact met Business Management meer gericht op construeren en minder op consulteren.

RISK IT kent drie aandachtsgebieden, die elk weer verdeeld zijn in drie aspecten. Ze lopen van de risico-awareness aan de top van het bedrijf tot operationele activiteiten op de werkvloer. Voor elk van de negen aspecten zijn doelstellingen geformuleerd. Om de doelstellingen te kunnen bereiken worden drieënveertig key-activities benoemd, dingen die je kunt doen om de doelstellingen te bereiken. In het kader van dit artikel gaat het te ver om elk van die processen tot in detail te beschrijven. We beperken ons tot hoofdlijnen en gaan daarna dieper in op processen die voor testers en testmanagers interessant zijn.

Het eerste aandachtgebied is Risk Governance. Anders dan men wellicht denkt, gaat het niet om de risico's zelf, nee, het gaat om hoe het topmanagement ermee om wil gaan.

Wil men een volger zijn die weinig risico loopt of een innovator met navenant meer risico en meer kans op grote winsten? Een communicatiemiddel is hier de Risk Appetite, met de risicokaart kaart waarop de onderkende risico's worden ingetekend. Later gaan we hierop in.



Het tweede aandachtgebied omvat onderzoek, en hier is natuurlijk testen te vinden. Van de Risk Evaluator wordt verwacht dat hij/zij data verzamelt (test), risico's analyseert (bevindingen proces), en het risicoprofiel bijhoudt op de risicokaart.

Ten slotte is er de situatie dat een risico optreedt en een maatregel moet worden genomen. Bijvoorbeeld: de backup moet teruggezet worden. De Risk Response persoon bereidt dat voor en test of de voorbereiding adequaat is, effectief en betrouwbaar. →

Risk Appetite

Het is heel interessant om eens te kijken naar hoe je met risico's om kunt gaan, zonder meteen over een concreet risico te praten. Dat doen we met de Risk Appetite, dat is de mate waarin een bedrijf risico's wil nemen om bedrijfsdoelstellingen te halen. Je hebt bedrijven die risico mijden (geneesmiddelenfabrikanten) en die risico's omarmen (Apple).

Het gebied 'Opportunity' heet zo, omdat de risico's hier eigenlijk er niet toe doen. Ze zijn klein qua omvang en treden zelden op. Als je al iets doet om deze risico's te mitigeren, zoals testen uitvoeren, kun je meteen besparen door het niet te doen.

De 'Acceptable' risico's zijn precies wat de naam zegt: het risico wordt (eventueel na genomen maatregelen) beschouwd als: we accepteren de verliezen als het optreedt. Als je hiervoor maatregelen wilt nemen (testen en herstellen is zo'n maatregel), dan is de opbrengst in de ogen van de Business minimaal.

De 'Unacceptable' risico's zijn in de ogen van topmanagement niet toelaatbaar: ze brengen de continuïteit van de onderneming teveel in gevaar. Maatregelen zijn nodig om het risico terug te brengen in het 'Acceptable' gebied. Maatregelen als testen, gericht op detecteren en herstellen. Of, en dat wordt vaak vergeten, men kan zich verzekeren tegen de slechte uitkomst of proberen de risico's te delen met partners.

De laatste categorie 'Really Unacceptable', dat zijn risico's, zo verschrikkelijk dat men dat risico absoluut niet wil lopen. Dus, men wil zekerheid dat het niet zal optreden, of men vermijdt deze risico's tegen (waarschijnlijk) hoge kosten. Dit is de TBWDWTFIP-categorie van bugs, een term van Derk-Jan de Groot: The Bugs You Don't Want To Find In Production.

Inspelen op Risk Appetite

Wij testers kennen al jaren het principe van Risk Based Testen (RBT). We kiezen RBT om onze beperkte resources (testtijd, mensen, omgevingen) zo nuttig mogelijk in te zetten. Daartoe maken we vooraf een inschatting van de risico's die we zien, bijvoorbeeld met een tweedimensionale PRIMA risicoanalyse. De uitkomst daarvan geeft een richtlijn voor de testinspanningen in het testplan.

PRIMA Product Risico Matrix		Project: Optimalisatie Bestuurs applicaties											
Kwaliteit top-10	Onderdeel ->	K2Burgerzaken	GWS	MVV	Corsa	K2Datadistributie	K2Financiën	GisVG	imPROMPTU	Architectuur, werkplekken	K2Onderwijs	BAG	K2Begraven
		100	13	13	13	13	12	8	8	8	4	3	3
1. Koppelbaarheid (S)	21	XX	X	XXX	XXX	X	XX	X				XXX	
2. Beschikbaarheid (S)	19	X	X		XX		X	X					
3. Performance (S)	18	X			X								
4. Herstelbaarheid (S)	13												
5. Foutbestendigheid (S)	10		X	X	X			X					
6. Volledigheid (P)	8	X					X		X				
7. Degradeerbaarheid (S)	3	X			X								
8. Beheerbaarheid (S)	3	X		X					X	X	X		X
9. Klantgerichtheid (P)	3												
10. Doorlooptijd (P)	2						X						



Als we terugdenken aan Risk Appetite zien we mogelijke verbeteringen. Waardoor we nóg beter met stakeholders communiceren.

Degradeerbaarheid, om er één te pakken, zou een 'Acceptable' risico kunnen zijn. Een test van of, en hoe, een systeem degradeert in een aantal essentiële scenario's vraagt veel inspanning, en is niet nodig voor een 'Acceptable' risico.

Wij testers herkennen Koppelbaarheid als risicovol kwaliteitsattribuut: interfaces werken vaak niet goed. Herstel kost meer tijd dan iedereen verwacht. Toch is het vaak lastig om voldoende middelen los te krijgen op basis van een risico-inschatting.

In de ogen van het topmanagement kan een niet-te-koppelen systeem makkelijk in de categorie 'Unacceptable' vallen. Door onze ervaring naast de Risk Appetite van het topmanagement te zetten kunnen we wellicht meer resources inzetten en zo sneller tot een werkend systeem geraken.

Wat in RBT niet meegenomen wordt, is potentieel verlies aan omzet en winst. Een langdurige test kan aan gemiste omzet veel meer kosten dan de test opbrengt. Zoals een keer in de telecombranche: in productie gaan met deze fouten kost € 112 miljoen, niet in productie gaan kost € 250 miljoen! Dus gaan we in productie. Door langszij te komen met Risk Appetite kun je zowel de verliezen in de Business beperken, als meer resources verkrijgen en daardoor uiteindelijk een beter testresultaat neerzetten.

Langszij komen bij Risk Appetite

Hoe bepaal je de Risk Appetite van een onderneming? Dat is niet voor de hand liggend. De Risk Appetite is iets van de ondernemingstop, en wij zijn opgegroeid in IT. Daar weten we het meest van af, toch?

De aanpak noemen we 'Langszij Komen'. Zoals twee schepen alleen goed contact kunnen maken als ze langszij liggen. Dat lukt als we de belangen van de Business kennen en erop kunnen inspelen. Wat voorwerk is nodig, maar de resultaten zijn de moeite waard. Als hulp bij het Langszij Komen, zijn er tegenwoordig beleidsstukken waarin iets over de Risk Appetite wordt gezegd. Dat heeft alles te maken met de crisis die ontstond toen banken te veel risico's namen. Het is de kunst vanuit deze stukken (die gaan over risico's in het bedrijf zelf) de vertaling te maken naar hoe men IT risico's zou indelen op dezelfde risicokaart.



Waar we ook eens naar kunnen kijken is, hoe de onderneming in het verleden met risico's omging, welke keuzen men maakte. Met alle drie onderzoeken als basis kunnen testers langszij komen bij het Business Management en de IT-risico's in de risicokaart plaatsen, naast de al bekende bedrijfsrisico's. Zodoende krijgt het management van de Business gevoel bij het belang van testen, voor henzelf.

		Wel testen bedrag in €	Niet testen bedrag in €
Testinspanning		€ -7.000	€ -
Testomgeving		€ -2.000	€ -
Overhead		€ -1.000	€ -
Gemiste opbrengst		€ -25.000	
Totaal kosten		€ -10.000	€ -
Herstelkosten		€ -1.250	€ -15.000
Blootgesteld risico	€ 50.000		€ -50.000
Netto rendement van test		€ -36.250	€ -65.000

Aanvullend kan het veel helpen Kosten en Opbrengsten van testen paraat te hebben, inclusief Business componenten. Een eenvoudig spreadsheet geeft, met name aan Business Managers met winstverantwoordelijkheid, inzicht in hun eigen referentiekader. Wij hebben daar regelmatig succes mee als andere manieren niet lukken.



Samenvatting

Tegenwoordig verwachten stakeholders dat er goedkoper en efficiënter getest wordt. Dat komt mede omdat zij het inzicht ontberen in de opbrengsten van testen.

RISK IT biedt nieuwe gezichtspunten om Risico Gebaseerd Testen op een hoger plan te brengen. Waar tot nu toe een risicoanalyse behulpzaam is bij het verdelen van testinspanning zal het inspelen op Risk Appetite bij de onderneming helpen testinspanningen te rechtvaardigen.

Tegelijk biedt RISK IT van ISACA handvatten om via risicomanagement op een heel andere manier, namelijk door Langsij te Komen, voor testen begrip te krijgen bij Business Management, management dat verantwoordelijk is voor omzet en winst. En dat testen eerder als een kostenpost zag, niet als investering.

Uiteindelijk zal RISK IT ervoor zorgen dat professioneel testen algemeen beschouwd wordt als een waardevolle bijdrage aan het bedrijfsproces. ←

TESTEN GROEIT MEE IN DE WERELD VAN ICT (VISIONAIR OF VOLGERS)

Door Heini Veneberg • h.veneberg@cimsolutions.nl



Wat is de toekomst van testen?

Als tester en testmanager heb ik me altijd de volgende vragen gesteld:

1. Hoe ga ik verder in dit vak?
2. Wat zijn de kansen en bedreigingen?
3. Waar staan we over vijf tot tien jaar?

In eerste instantie zou je kunnen denken: waarom zover vooruit kijken?

De meest gehoorde reactie op deze vragen is: 'Ik vind het al moeilijk genoeg om één of twee jaar vooruit te kijken'.

Toch is het belangrijk je keuzes niet alleen op de korte termijn te richten, maar ook de langetermijngevolgen proberen in te schatten. Daarbij is het van belang om veranderingen in het ICT-landschap te zien. Door deze veranderingen zijn er bedreigingen voor de testdisciplines, maar ook kansen.

In mijn huidige functie begeleid ik medewerkers in hun persoonlijke ontwikkeling. Vaak hoor je de uitspraak: 'Ik wil een testspecialist worden.' De vraag is alleen 'wanneer ben je een specialist?'. Je bent een specialist als je ergens veel verstand van hebt. Als je dat heel plat zegt, ben je iemand die een bepaald kunstje heel goed kan. Niks mis mee, zou ik zo zeggen. Maar ik voeg er meestal iets aan toe bij het begrip specialist. Hoe weet je omgeving dat je de specialist bent en wat doe je eraan om specialist te blijven? Met andere woorden, ben je een visionair of een volger? In mijn ogen een belangrijke vraag die elke persoon zich zou moeten stellen. Waarbij ik direct wil aangeven geen waardeoordeel te hebben met het zijn van visionair of volger.

Alleen als iemand keuzes maakt voor aspecten in zijn vak, dan kan dat grote gevolgen hebben voor zijn toekomst in het vak. Specialisaties kunnen uitsterven. Als specialist moet je op tijd inzien, dat je jouw specialisatie moet bijsturen. Met andere woorden, je moet zorgen dat je niet een uitstervend ras wordt.

Het testvak is de afgelopen jaren volwassen geworden en geaccepteerd als een onmisbare schakel in het ICT-vak. Dat is natuurlijk goed. Nu op onze lauweren rusten qua acceptatie, is het begin van uitsterven van het testvak. ICT-systemen worden complexer, dus zou je denken dat testen steeds belangrijker zal worden.

Met name het eerste deel is helemaal waar. Door het complexer worden van ICT-systemen, kost het testen van systemen ook steeds meer. Doordat het complexer wordt, kun je als testdiscipline minder goed aantonen dat het systeem helemaal correct werkt. In de praktijk komen cruciale fouten in productie regelmatig aan het licht, die zelfs het landelijk nieuws halen. Denk aan:

1. Het niet waterdicht zijn van informatiesystemen;
2. Dagenlang niet kunnen bellen of mobiel internet gebruiken;
3. Versturen van brieven naar overleden personen.

→



Oplossingen

Het ICT-vak ontwikkelt zich verder om oplossingen te bieden om oplossingen voor complexe vraagstukken te implementeren, maar ook om antwoorden te geven op de vraag naar het goedkoper realiseren van ICT-systemen. Dit kan zorgen dat men de testdiscipline niet meer nodig heeft, oftewel de testdiscipline kan hierdoor uitsterven.

De afgelopen vierentwintig jaar heb ik me regelmatig de vraag gesteld: wat ga ik over vijf tot tien jaar qua keuze tussen visionair en volger doen? Ben ik een visionair? Zelf vind ik dit een lastige vraag. Persoonlijk zie ik me als een volger, die zijn visie over het testen regelmatig geeft. Of anders gezegd, ik sta open voor de veranderingen om me heen, probeer in te schatten wat de gevolgen zijn voor mijn vak. Waarom doe ik dit? Om dit uit te leggen gebruik ik een analogie met de textielindustrie in Twente.

De textiel in Twente is in de afgelopen decennia gigantisch afgenomen. Mijn grootouders werkten in de textiel, bij fameuze namen zoals Ten Cate, Jansen & Tilanus.

Jansen & Tilanus begonnen in 1869 met een stoomweverij in het Twentse Vriezenveen en er kwam in 1874 een tricotagefabriek bij. Het bedrijf was lang succesvol. In de jaren zestig echter moest de onderneming inkrimpen en, zoals zoveel andere Nederlandse textielfabrikanten, begin jaren tachtig haar deuren voorgoed sluiten. Ten Cate daarentegen bestaat nog steeds als bedrijf, toonaangevend op het gebied van innovatie in de textiel. De groot-schalige productie is geoutsourcet naar goedkope landen.

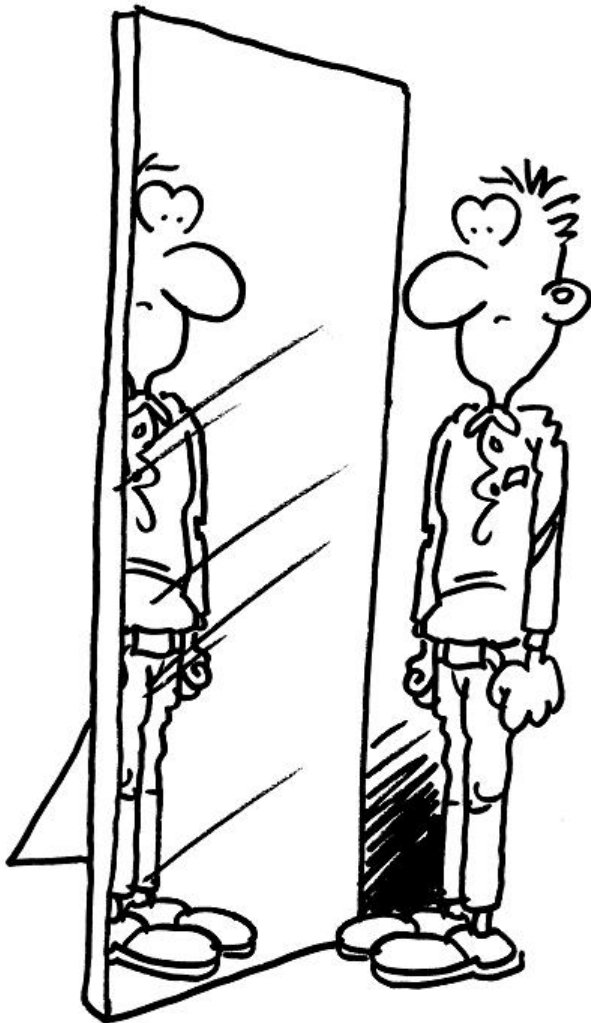
Deze leidraad van de textiel is altijd bepalend geweest voor mijn keuzes in mijn carrière. Ik zie veel overeenkomsten in dit traject. In de volgende tabel heb ik textiel en ICT naast elkaar gezet.

Onderwerp	Textiel	ICT
Complexiteit toename	Meer keuze in diversiteit van kleding	Systemen worden meer en meer aan elkaar gekoppeld
Uitbesteden	Productie is verplaatst naar goedkopere landen	ICT-ontwikkeling en testen worden uitbesteed naar Oost-Europa en/of Azië
Innovatie	Productie van beschermende kleding, met als doel betere bescherming en toch heel goed draagbaar	Cloud computing Model Based Testing

Platformen zoals TestNet hebben mij de afgelopen zeven jaar geholpen antwoorden te vinden over veranderingen in het testvak. Als tester ben je daarmee niet afhankelijk van je eigen ervaringen alleen, maar je kunt ook zien welke veranderingen er om je heen gebeuren. Dit verrijkt je kennis over je vak en zorgt ervoor dat je een eigen visie kunt ontwikkelen zodat je zelf groeit van volger naar visionair.

Kansen

Op dit moment bekijk ik zaken als de praktische toepassing van Model Based Testing en hoe je als tester antwoord geeft op vragen over cloud toepassingen. Met name cloud toepassingen bieden kansen voor testers. Hoe krijg je het geheel beheersbaar qua testkosten en biedt de cloud voldoende testkwaliteit? Persoonlijk denk ik dat aggregatie van testresultaten een oplossing kan zijn voor testen van cloud toepassingen. Uit ervaring heb ik gemerkt dat we op elk niveau krampachtig alles proberen te testen. Door het toepassen van goede entry criteria en door middel van het 'gate process' kunnen we vaststellen wat de kwaliteit is van het opgeleverde product. Vervolgens moet de organisatie ook een afkeur accepteren op een 'gate'. →



Als dit proces goed gevolgd wordt in een organisatie, dan zal de aggregatie van testresultaten ervoor zorgen dat de testkosten beheersbaar blijven. Echter, niet alleen de organisatie bij acceptatie, maar ook de tester bij zijn eigen werkzaamheden moet leren vertrouwen op een eerder opgeleverd testresultaat. Dit is een kans voor de testdiscipline om de regie te nemen op gebied van aggregatie van testresultaten. Dit vraagt andere expertise van een tester, testcoördinator en testmanager. Hoe dwing je respect af in de keten ten aanzien van het opgeleverde testresultaat?

Daarom: kijk als tester in de spiegel en stel je regelmatig de vraag: 'Wat ben ik voor een tester: volger of visionair?'



SOFTWAREAPPLICATIES: GOED GEBOUWD EN TOCH NIET AF!?

Door Jurgen van Amerongen • Jurgen.van.Amerongen@bartosz.nl



Productgereed versus productiegereed

Veel softwareapplicaties eindigen op de plank en worden nooit gebruikt. Wie herkent dit niet? Zonde van de tijd, de inspanning en het geld. Weer een illusie armer en een frustratie rijker. Oorzaak: vaak is softwareontwikkeling is een doel op zich zelf geworden met de nadruk op techniek. 'State of the art' lijkt zelfs belangrijker te zijn dan functionaliteit. Het gevolg van deze insteek is dat een softwareapplicatie niet in productie kan worden genomen omdat de organisatie niet of nauwelijks is voorbereid.

De processen die moeten worden ondersteund krijgen onvoldoende aandacht en sluiten dus niet aan op de nieuwe softwareapplicatie. De softwareapplicatie is weliswaar klaar (productgereed), maar dus nog niet productiegereed. Succesvollere projecten stellen het proces centraal. Van daaruit wordt gekeken, hoe dit proces geautomatiseerd kan worden ondersteund. Vervolgens wordt vastgesteld wat dit betekent voor de gebruikersorganisatie. Dit artikel presenteert een integrale aanpak waarmee de gebruikersorganisatie optimaal wordt voorbereid op het proces en de bijbehorende ondersteunende softwareapplicatie. Hiermee wordt de kans op succes aanzienlijk vergroot.

De praktijk

Te veel ICT-projecten mislukken. Hoewel de meningen nogal verdeeld zijn over de percentages, blijkt uit diverse marktonderzoeken dat een significant aantal projecten anders verloopt dan verwacht en niet tot het gewenste resultaat leidt. Dit geldt voor onder andere de doorlooptijd, de uiteindelijke kosten, de scope of de kwaliteit van het product of de dienst. We kennen allemaal wel voorbeelden waarbij tijdens het project aanvullende eisen werden gesteld of bestaande eisen werden herzien. Onduidelijke eisen leiden tot grote verwarring en verkeerde aannames of interpretaties. Het uiteindelijke product of dienst voldoet niet aan de vooraf gestelde eisen of verwachtingen. Daarnaast komt het veelvuldig voor dat het project uitloopt in de tijd of het budget ver overschrijdt. Dit wordt niet altijd als een mislukking gezien, maar goedgepraat als een aanpassing op basis van voortschrijdend inzicht. Hoe dan ook, het is een afwijking van wat oorspronkelijk was bedacht en verwacht.

Dat dit fenomeen niet onopgemerkt is gebleven, blijkt uit de grote hoeveelheid literatuur waarin allerlei oplossingen worden geboden voor alle denkbare oorzaken. Dit varieert van een projectmanagement aanpak tot voorstellen voor communicatieverbetering. Ook worden er tools aangeboden om het project te ondersteunen. Ondanks alle mogelijke maatregelen zullen projecten blijven falen. Zeker zolang de focus blijft gericht op de softwareapplicatie waarbij de techniek vaak vooropstaat.

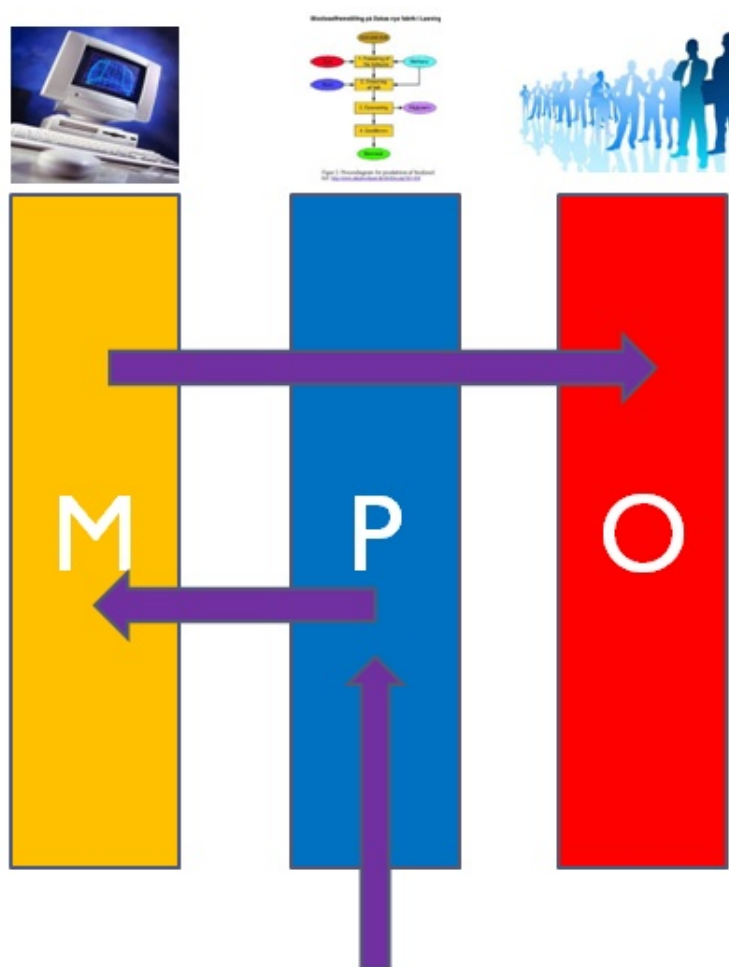
Introductie productiegereed

Met softwareapplicaties ontwikkelen alleen is men er nog niet. Vooropgesteld dat de softwareontwikkeling goed is verlopen en het ontwikkelde product klaar is, blijft de vraag of het dan ook productiegereed is. Hier ligt een wezenlijke en misschien wel de belangrijkste oorzaak van het falen van projecten. Alleen aandacht voor softwareapplicatie (productgereed) is namelijk niet genoeg. Voor 'productiegereed' moet softwareapplicatieontwikkeling integraal samen met twee andere relevante aandachtsgebieden worden beschouwd namelijk: processen en organisatie. Op deze wijze wordt gewaarborgd dat niet alleen naar de nieuwe functionaliteit wordt gekeken, maar ook naar de omgeving waarin deze wordt ingezet. →

Door hier vroegtijdig op te anticiperen en de nodige maatregelen te treffen wordt bereikt dat niet alleen de applicatie gereed is maar dat deze ook daadwerkelijk kan worden gebruikt door bijvoorbeeld goed opgeleide gebruikers.

Procesgestuurde aanpak

Een aanpak die in bovengenoemde integraliteit voorziet, is de procesgestuurde aanpak. Deze stelt de processen centraal. Het maakt niet uit of het om nieuwe of om bestaande processen gaat. Voor de toekomstige, gewenste situatie moet eerst helder zijn hoe het proces eruitziet. Dit kan een nieuw proces zijn, maar ook een bestaand proces dat al dan niet wordt aangepast. Dit proces moet tot op functieniveau zijn gemodelleerd. Vervolgens kan worden bepaald hoe deze functies geautomatiseerd kunnen worden ondersteund. Dit levert een pakket van eisen op, vertaald in specificaties. Zodra dit helder is, kan worden bepaald wat dit betekent voor de organisatie. Denk hierbij bijvoorbeeld aan opleidingen, gebruikersprofielen en autorisaties.



Figuur 1. Procesgerichte aanpak.

Deze figuur laat zien dat het proces (P) centraal staat. Op basis van dit proces wordt vastgesteld welke middelen (M) zoals softwareapplicaties het proces ondersteunen. Aan de hand van P en M kan dan worden vastgesteld welke veranderkundige gevolgen dit heeft voor de organisatie (O).

De procesgerichte aanpak richt zich niet alleen op het primaire proces, maar ook op de secundaire processen. Met het primaire proces wordt bedoeld de kernfunctionaliteit zoals klantenregistratie. Secundaire processen zijn nodig om het primaire proces te kunnen uitvoeren en beheersen. Denk hierbij aan beheerprocessen en productiebesturingsprocessen, maar ook aan beveiliging en performance. Vanuit al deze processen kunnen eisen worden gesteld aan de te bouwen softwareapplicatie. Het zal niet de eerste keer zijn dat een systeem is gebouwd dat precies doet wat het moet doen, maar dat niet vooruit te branden is of het aantal gebruikers niet aankan.

Een andere manier om naar de volledigheid van de processen te kijken is vanuit de levenscyclus van een systeem. Hierbij wordt niet alleen de vraag gesteld wat er nodig is om een systeem in de lucht te krijgen, maar ook wat er nodig is om het systeem in de lucht te houden. Met andere woorden, wat is er nodig om de continuïteit te waarborgen? →

Waarom deze aanpak?

Waarom zou deze aanpak tot meer succes leiden? Het antwoord hierop is gelegen in een combinatie van factoren. De belangrijkste factor is de scope. Door niet alleen naar de softwareapplicatie te kijken, maar ook naar organisatie en processen en naar de samenhang tussen deze aandachtsgebieden, worden alle aspecten meegenomen die relevant zijn voor het welslagen van een ICT-project. Hiermee is de volledigheid gewaarborgd. Een andere factor is het centraal stellen van het proces. Er ontstaat focus op alleen datgene wat strikt noodzakelijk is om te bouwen. Vanuit de processen kunnen de eisen worden gedefinieerd voor het systeem. Bij het centraal stellen van de softwareapplicatie is de verleiding groot om er van alles bij te verzinnen wat handig zou kunnen, de zogenaamde 'nice-to-have' of de categorie 'misschien wel nodig'. Onnodige ballast die alleen maar leidt tot verhoging van de complexiteit en daarmee extra vertraging en kosten. Een derde belangrijke factor is de betrokkenheid van de business. Deze aanpak is alleen maar goed uitvoerbaar als de business hier vroegtijdig en met de juiste mensen bij betrokken is. Een bijzonder gunstige bijkomstigheid is de acceptatie van het nieuwe systeem. Door de business erbij te betrekken wordt zij actief deelgenoot van het project waardoor eventuele weerstand zal afnemen. Dit komt de implementatie ten goede. De laatste factor die ik wil noemen is de planning. Doordat alle aspecten vroegtijdig worden onderkend en door de impact hiervan nauwkeurig te bepalen, kan een realistische planning worden opgesteld. Realistisch in de zin van compleetheid waarmee onaangename verrassingen kunnen worden voorkomen.

Ter illustratie

Laten we een herkenbaar voorbeeld nemen om deze aanpak te illustreren. Een metafoor die tot de verbeelding moet spreken is het containervervoer. Stel dat dit nu plaatsvindt over de weg met grote opleggers. De directie van een fictieve transportorganisatie wil de capaciteit verdubbelen, maar ook kostenefficiënter gaan werken. Uiteindelijk wordt besloten om de containers over het water te gaan transporteren met containerschepen. Het hoofdproces is en blijft containervervoer. Echter, de wijze waarop hieraan invulling gaat worden gegeven raakt vrijwel alle onderliggende processen. Om te beginnen het transportproces: een containerschip in plaats van een oplegger. Dit heeft consequenties voor onder andere de besturing, het beladen, het onderhoud, de planning et cetera. Je zult begrijpen dat, als het schip wordt opgeleverd en met de nodige feestelijkheden te water wordt gelaten, er geen container kan worden verscheept zolang heel veel andere zaken niet geregeld zijn. Door alle onderliggende processen tegen het licht van de gemaakte keuze te houden worden de gevolgen en daarmee de te nemen maatregelen voor de organisatie duidelijk.

Aandachtspunten

De procesgestuurde aanpak stelt hoge eisen aan de gebruikersorganisatie. Anticiperen op komende veranderingen betekent vroegtijdig handelen. De praktijk leert dat de bemensing van een gebruikersorganisatie is afgestemd op de kerntaak productie. Extra capaciteit om naast de kerntaak een omvangrijk project te ondersteunen door alvast voorbereidingen te treffen is er nauwelijks. Meestal wordt dan ook afgewacht tot men er niet meer onderuit kan. Helaas gaat hiermee kostbare tijd verloren. Ook eventuele fouten worden door de afwachtende houding (te) laat ontdekt waardoor de herstellkosten exponentieel toenemen. Geadviseerd wordt om bijvoorbeeld vroegtijdig een implementatiemanager te benoemen om de organisatie klaar te stomen voor de nieuwe processen en de bijbehorende middelen. →

Samenvatting

Veelal wordt de ontwikkeling van een softwareapplicatie onvoldoende in samenhang met de te ondersteunen processen en de gebruikersorganisatie beschouwd. Het product 'softwareapplicatie' is dan weliswaar gereed maar kan nog niet in productie worden genomen ofwel het is nog niet productiegereed. Bij een procesgerichte aanpak worden de drie aandachtsgebieden processen, softwareapplicatie en gebruikersorganisatie integraal beschouwd. Door de processen centraal te stellen en als vertrekpunt te nemen, zal de softwareapplicatie hierop beter aansluiten en behoudt men focus. De gebruikersorganisatie kan zich tijdig op de verandering voorbereiden. Dit bevordert de implementatie en vergroot de acceptatie.

Bronnen

Anon. (2006). *Zwijgen grootste oorzaak mislukte projecten*. Managersonline.nl

Roos, J. (2010). *Wijkt de ict-sector af bij 'mislukte projecten'?* Computable.nl

Lindhout, K. (2010). *5 Valkuilen bij veranderen IT*. FM.nl

Marselis, H.J. et al. (2007). *TestGrip*. LogicaCMG

Oosterhout, B. van (2007). *Het ict-drama: Hoe de overheid miljarden verspilt*. Intermediair.nl

Rooyen, J. van, et al. (2011). *Project de Baas*. Tutein Nolthenius



IOS APP TESTEN

Door Rudi Niemeijer • rudiniemeijer@gmail.com



Apple

Apple heeft met de iPhone bestaande concepten (aanraakscherm, computer op zakformaat, batterijgevoed volwaardig besturingssysteem, koppelingen met WiFi, 3G, bluetooth, en GPS, foto/videografie in een telefoon, gratis SDK) die voorheen slechts als uitzondering beschikbaar waren gebundeld, tot kunst verheven en aan het grote publiek beschikbaar gemaakt.

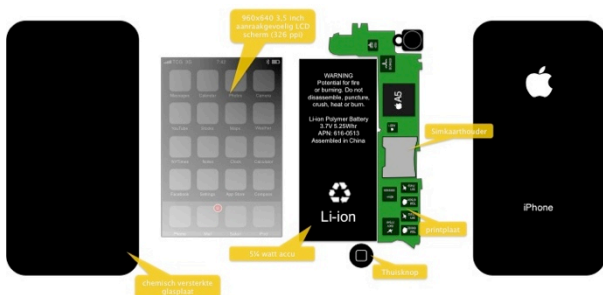
Toen Apple in begin 2007 bekendmaakte dat ze van plan waren wel tien miljoen exemplaren in het eerste jaar te verkopen geloofde niemand dat ze het konden waarmaken: veel te duur, veel te groot, geen ervaring met telefoons, een gadget voor Fanboys, geen fysiek toetsenbord, geen ervaring met operators en geen applicaties. Nokia en Motorola, de marktleiders, zouden er geen seconde van wakker liggen. In 2012 worden er meer dan 33 miljoen



iPhones per kwartaal (150 miljoen per jaar) verkocht en zien de meeste smartphones er uit als een regelrechte iPhone kloon, voorzien van een 3,5" kleurenscherm en een aanraakscherm zonder fysiek toetsenbord. Android en iOS zijn de meest gebruikte besturingssystemen, Apple en Samsung de grootste telefoonmakers en Nokia en Motorola lopen miljarden omzet mis. Ook de sprong naar de 'tablet' in 2010 werd niet met gejuich ontvangen: een groot uitgevallen iPhone en geen reden om de laptop te vervangen. Toch worden er zo'n 50 miljoen iPads per jaar verkocht, waarbij de verkochte aantallen nog ieder jaar verdubbelen. Ook de Android tablets doen het goed, met Amazon en Samsung als grootste spelers op deze markt.

Een App maken

Het goede nieuws voor de consument is eenvoudig bruikbare en zeer betaalbare computerkracht met een keur aan eenvoudig te selecteren 'Apps'. Door de vergroting van de doelgroep (iPads doen het goed bij consumenten van 2 tot 102 jaar), de onvoorstelbare omvang van het klantenbestand (200 miljoen actieve creditcards telt de Apple App Store), de snelheid waarmee een nieuwe App wordt verkocht en vooral de laagdrempeligheid van het ontwikkelen en aanbieden van een App (koop een Mac, meld je aan als developer, schrijf de App, kies een verkoopprijs

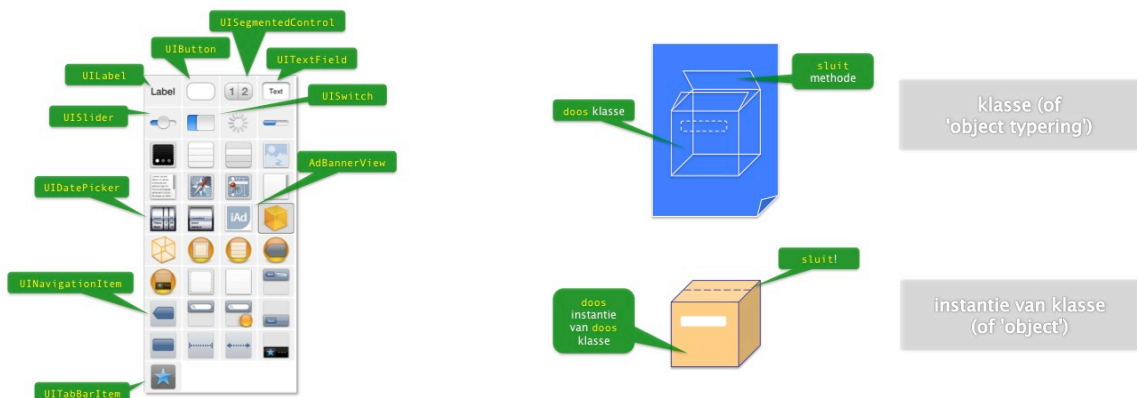


en publiceer 'm) zorgen ervoor dat het aantal Apps in 2012 hard op weg is richting de 1 *miljard* (het totaal aantal verkochte Apps ligt daar al ruim voorbij). Veel scholieren hebben met het maken van een eenvoudige App in enkele maanden hun studietoelafbetaald en je ziet het steeds meer gebeuren dat een goed idee voor een App niet zozeer uit de ICT, maar juist vanuit de materiedeskundigheid ontstaat. →

Er is geen zicht op het uitdoven van de belangstelling voor iOS (dit geldt evengoed voor Android, hoewel dit met segmentering van de hardwaremarkt nog wel een harde noot te kraken heeft) en de verkopen van Apple en Samsung alleen zijn nu al goed voor een deel van de pc- en laptopmarkt. De toekomst valt niet te voorspellen, maar het is een steeds meer voorkomende lijn van redeneren dat de tabletmarkt, die nog voor een groot deel door de iPad wordt ingenomen, op termijn een groot deel van de inkomsten van de pc-markt 'opsnoept'. Met de trend dat veel mensen liever een laptop in huis hebben dan een 'grote pc' resulteert dit op termijn mogelijk in het wegvallen van de behoefte aan pc's, de opkomst van kwalitatief goede laptops en meer dan één tablet in ieder huishouden.

Betekenis voor testen

Wat betekent dit voor ons testers? In elk geval betekent dit dat de kans groot wordt dat we met iOS en Android te maken krijgen als 'nieuw' besturingssysteem. Ook hebben we te maken met de 'App Store' effecten (ook al zou een App binnen de eigen organisatie worden uitgerold op bedrijfs-iPads) die onze testaanpak beïnvloeden. De gebruikers van de App Store (en dat zijn alle gebruikers van een iPad) weten hoe een App eruit zou moeten zien omdat ze deze in honderdvoud dagelijks bekijken. Door de ervaring wijs geworden weten ze dat een App het altijd doet (en zo niet, dan is er wel een andere App die het wel doet en door negatieve publiciteit is die niet werkende App dan zo uit de App Store verdwenen) en dat een App er te gek uitziet en één ding vooral heel goed doet. Door dat kwaliteitsgevoel wordt ook de Enterprise App beïnvloed en dat is iets waar we als tester rekening mee moeten houden. Doordat de lat voor het ontwikkelen laag is en de doorlooptijden relatief kort zijn, is het zaak om zo snel mogelijk als tester 'bij te springen'. Dit betekent delen van de App testen, gebruikmaken van de iPhone en iPad Simulator, met Instruments de werking van de App monitoren en met een selecte groep van acceptatietesters de algehele App-beleving in detail bemonsteren. Dat alles bevat veel activiteiten die we normaal ook doen, maar zal voor veel testers toch wennen zijn. Immers, hoe werkt Xcode, hoe zit Cocoa Touch in elkaar? Wat zijn de succesfactoren van een App? Waar moet je op letten bij het testen? Heeft een grenswaardenanalyse nog wel zin? Of is het toch beter om een procescyclustest uit te voeren? →



Extra kennis benodigd

Het is uiterst zinvol om als tester meer te weten te komen over het (iOS) platform en de ontwikkelmethoden en tools hiervoor. En het gebruik van een Macintosh, want daar vindt de ontwikkeling van iOS Apps op plaats. Het kan helemaal geen kwaad om een paar dagen op cursus te gaan om te leren een App te ontwikkelen.

Immers, als tester weet je dan precies wat een ontwikkelaar moet doen, zoals object-georiënteerd ontwerpen en programmeren. Andere kennisonderwerpen zijn: de gebruikersinterface die op 'vingergebaren' is gebaseerd, een Unix besturingssysteem, een onvoorstelbare connectiviteit (internet, locatievoorzieningen, databases, webservices, muziek, video, foto's, webpagina's) die standaard onderdeel van de API's is. Er is heel wat te leren.

Complete werkomgeving

Apple levert een summum aan informatie over alles wat nodig is om te ontwikkelen en te testen voor iOS, waartoe (gratis) tools, instructievideo's, documentatie, App voorbeelden, Cocoa frameworks, object-georiënteerd programmeren en nog veel meer behoren. De website developer.apple.com/ios is het startpunt hiervoor en vormt voor velen de kennismaking met een volledig nieuwe carrière als iOS App developer of tester. ←



TESTEN VAN DE API ACHTER EEN MOBILE APP

Door Marc van 't Veer • marc.vantveer@polteq.com



'There's an app for that' is een officiële slogan en commercial van Apple die direct begrepen wordt door de meeste mensen. Ongeacht welke functionaliteit gewenst is: er kan een app voor gevonden worden. De marketingafdeling van T-Mobile denkt precies zo en wil klanten direct toegang geven tot hun essentiële data zoals facturen en belstatus. Maar hoe geef je deze ervaring aan iedereen en niet alleen aan de iPhone klanten? Wat doe je dan voor je klanten met een BlackBerry, Windows of Android telefoon? 'En we willen ook nog een Facebook integratie!' voegde een marketing medewerker eraan toe.

Met deze diversiteit aan vragen heb je een API nodig! Maar wat is de toegevoegde waarde van een systeemtester in een cloud enabling project? Waarom niet direct een app met een API verbinden en zo testen?

Screen scrapers

Op dit moment bestaan er zogeheten screen-scrapers die de T-Mobile website scannen, de daar gevonden data interpreteren en presenteren. Als website verandert, dan valt een app om. De interpretatie is niet altijd correct en het wachtwoord van T-Mobile wordt opnieuw gebruikt in de app. Als antwoord hierop wil T-Mobile een beveiligde toegang geven en controle krijgen over de data die wordt uitgewisseld door de introductie van een API.

Wat is een API?

Voordat je een API kunt (systeem)testen, moet eerst begrepen worden wat het is. Een definitie van een API kan gevonden worden op Wikipedia: een *Application Programming Interface*. Dit is een verzameling definities op basis waarvan een [computerprogramma](#) kan communiceren met een ander programma of onderdeel. Hier is een API een publieke webservice op internet. Deze API definieert een generieke toegang tot de functionaliteit daarachter. In het kort is een API een raamwerk voor de communicatie tussen systemen. Voor T-Mobile is de API een zelfstandige interface en is voortgebouwd op de interne Service Oriented Architecture (SOA). Hiermee wordt de keten een stap groter. Vanuit systeemtest gezien, zit in de korte API-definitie alles:

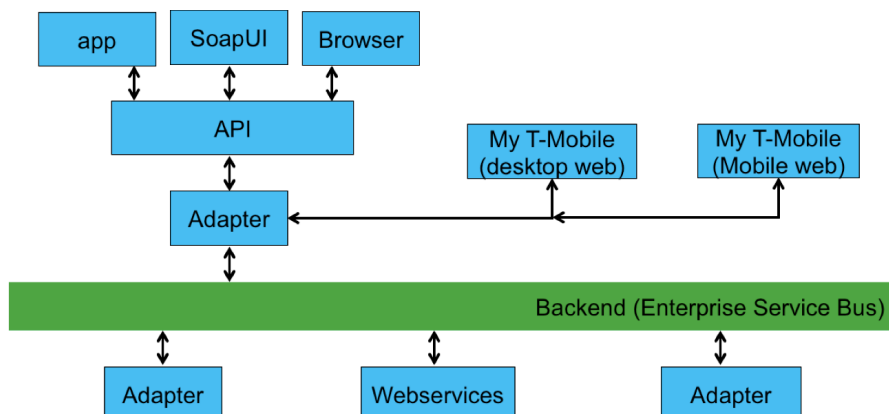
- het raamwerk en alle elementen hierin;
- de communicatie;
- de data die verzonden wordt en gepresenteerd;
- een 'tussensysteem' dat aangeeft dat er een integratie is.

Systeemtest van een API en SOA

Zo bekeken bestaat de systeemtest uit drie onderdelen: testen van het raamwerk (de structuur), testen van de data die gecommuniceerd wordt (functionaliteit) en testen van de integratie met andere systemen. Een eerste belangrijke stap is te begrijpen hoe een API bediend kan (moet) worden. De API is een publieke service die bijvoorbeeld via een URL in een browser aan te spreken is. De URL bevat het adres van de service en het uit te voeren commando. Een voorbeeld van een publieke API van de openbaar vervoerder Nederlandse Spoorwegen (NS): <http://nsapi.xmpp.openov.nl/stations>. Dit is het adres voor alle NS stations in Nederland, als je hierachter stations typt, dan krijg je de stations namen en met stations/ (met forward slash) de afkortingen om verder te navigeren.

→

Het is te vergelijken met de Windows command line. Er is geen front-end aanwezig alleen cryptische commando's en op basis van de output navigeer je verder.



Schematisch overzicht uitbreiding SOA architectuur met API

Risico's

Een typisch risico voor het testen van een API is dat toekomstige gebruikers onbekend zijn en waarschijnlijk nog niet bestaan. Een ander risico is dat integraties tussen bijvoorbeeld app en API nog niet zijn uitgevoerd. De toekomst is nog niet gespecificeerd, hoe kun je hier dan testen voor opstellen en uitvoeren? Andere voorbeelden waarbij risico's optreden zijn (test)data en scope. Met welke data kan een toekomstige gebruiker tegen de API 'praten'? Waar begint de verantwoordelijkheid van een tester en waar eindigt deze als je de API moet testen?



My T-Mobile app met testdata van de API.

Acht fasen

Bij het realiseren, bouwen, van software doorloopt T-Mobile normaal vier fasen, van development tot en met automated regression test. Bijzonder bij de introductie van de API zijn de extra integratiefasen, zoals productie-integratietests. De strategie voor het testen van de API bij T-Mobile bestaat uit: 'Het gecontroleerd integreren in de complete infrastructuur (SOA-architectuur)'. →

In totaal worden er acht fasen doorlopen:

1. Development, Systeemtest en Systeem Integratie Test (SIT) door T-Mobile;
2. Integration (internal: API and complete backend);
3. app Development door app leverancier;
4. Acceptance;
5. Integration (external: API with prototype app);
6. Automated regression test;
7. Productie-integratietest;
8. Aftercare.

De zevende fase, productie-integratietest, bestaat uit meerdere stappen, waaronder geselecteerde T-Mobile medewerkers die op productie als eerste de app kunnen downloaden en testen of de complete keten werkt. De aanpak van systeemtest wordt gestructureerd door de REST architectuur van de API. (Voor informatie over REST zie: <http://nl.wikipedia.org/wiki/REST> en <https://capi.t-mobile.nl/documentation/rest>.) De systeemtest heeft vooral een technische insteek en niet zozeer een functionele. Onderdelen van de systeemtest zijn bijvoorbeeld het testen van de resources, internet media types, de caching van output van een API-aanroep en de verschillende presentatie vormen (HTML, XML en JSON).

(Informatie over internet media types, zie: http://en.wikipedia.org/wiki/Internet_media_type.)

Beperkingen testomgeving

De systeemtest is een gecontroleerde fase in een gecontroleerde omgeving. De meeste defects vind je op het grensvlak tussen de API en de app. Zo bleek de app niet om te kunnen gaan met de standaard HTTP status codes, zoals 401 - Unauthorized. Een reden voor deze melding kan zijn dat er een 'Invalid username or token' is, maar ook 'Account has been temporarily locked out'. Welke melding moet de app nu laten zien? Een oplossing is een implementatie van een unieke API Error Code voor elke situatie. Dit is slechts één voorbeeld van een integratiedefect (tussen app en API). Een belangrijke les is dat een aantal tests niet uitgevoerd kan worden op een testomgeving. Dat komt doordat op deze omgeving niet alle klant/product configuraties getest kunnen worden, waardoor mogelijke defects pas gevonden worden op de productie omgeving. Er is dus behoefte aan een testfase als 'Productie Tests'. Als een bepaald abonnement niet getoond wordt zoals verwacht, dan bestaat er voor de website een backup (trick) om dit te corrigeren in de presentatie, omdat deze door T-Mobile zelf beheerd wordt. De belangrijke les hieruit is dat voor de API geen controle is over de presentatie van de data op de app.

Conclusie

Na alle testfasen is de API succesvol live gegaan. De API is sinds december 2011 door 250.000 gebruikers aangeroepen via een app op een iPhone of Android telefoon. Verder is de API voorbereid voor een Windows en Blackberry telefoon, maar ook voor Facebook. Voor een systeemtester blijkt het bij het testen van een API een groot voordeel te zijn als je ervaring met SOA-omgevingen hebt. Als systeemtester heb je bovendien ook nieuwe verantwoordelijkheden bij alle fasen van het hele project, bijvoorbeeld voor het voorbereiden van testdata voor andere testfasen/testafdelingen en het uitvoeren van productietests.

Een API's is een enabling technologie die de deur opent naar de nieuwe wereld van cloud-computing! ←

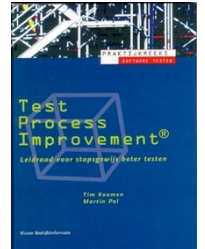
VAN TPI® NAAR TPI NEXT® - 15 JAAR

Door Loek Wilhelmus • loek.wilhelmus@sogeti.nl



Historie

Ook TPI (Test Process Improvement, Testproces verbetering) bestaat (bijna) vijftien jaar; het eerste TPI boek van Sogeti, toen nog Iquip geheten, is gepubliceerd in 1998; de ideeën daarvoor zijn in de loop van 1997 ontstaan. Sindsdien heeft TPI een stormachtige ontwikkeling doorgemaakt en wordt nu in alle werelddelen gebruikt. Van het oorspronkelijke boek zijn, in de verschillende talen zo'n 17.000 exemplaren verkocht en er zijn duizenden tevreden organisaties.



De basis van TPI

De oorspronkelijke gedachten voor TPI als methode voor verbetering van het testproces zijn ontstaan als gevolg van de beperkte aandacht die in (toen nog) CMM werd besteed aan het testproces. Alleen de entiteiten VAL (Validation) en VER (Verification) waren gericht op het testproces. Ook het ontbreken van de mogelijkheid van een stapsgewijze verbetering van het testproces heeft meegespeeld bij de gedachten om TPI te ontwikkelen. TPI is gebaseerd op een model dat de volwassenheid van het testproces beschrijft. Dit gebeurt zowel voor de huidige situatie als een gewenste, toekomstige situatie.

De evolutie van TPI

TPI is in de loop der jaren natuurlijk geëvolueerd en verbeterd; gebleven zijn de belangrijkste eigenschappen en kenmerken:

- verbeteringen worden stapsgewijs en beheersbaar doorgevoerd;
- praktijkgericht;
- objectief en controleerbaar;
- de mogelijkheid om keuzes en prioriteiten te stellen;
- zeer gedetailleerd;
- snel inzicht;
- onafhankelijk van methoden en technieken.

De boeken

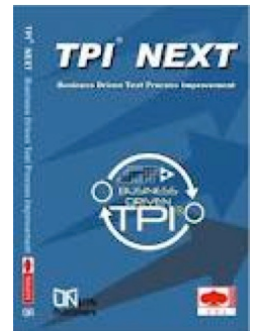
Het eerste boek is gepubliceerd in het Nederlands; daar kwam al snel een Engels versie bij. Die werd gevolgd door versies in het Duits, Chinees en Japans. Daarnaast is voor en in samenwerking met de Automotive industrie Duitsland een TPI Automotive ontwikkeld (TPI_A). TPI_A wordt in Duistland tot op de dag van vandaag nog veelvuldig toegepast. Ook zijn er varianten gepubliceerd over TPI, in de vorm van White Papers, zoals bijvoorbeeld een TPI-SOA versie. →



De opvolging

In de loop van 2008 is een project gestart om het model aan te passen aan recente ontwikkelingen in de ICT. De belangrijkste eisen waren:

- meer aandacht voor aansluiting met de andere SDLC (Software Development Life Cycle) processen zoals ontwerpen en bouwen;
- verbeterde presentatie van de volwassenheidsmatrix;
- scherpere definities van de aandachtsgebieden;
- aansluiting met ontwikkelingen als CMMi, Agile, outsourcing etc.;
- 'Business drivers' als uitgangspunt: de mogelijkheid om het model flexibel in te richten op specifieke business drivers als kostenbesparing, transparantie van het testproces en time-to-market.



Dit heeft in 2009 geresulteerd in een nieuw boek, TPI Next[®], geschreven door een internationaal team van auteurs. In eerste instantie is het boek in de Engelse taal geschreven; later is ook een Duitse versie gepubliceerd. De naam TPI Next is gekozen omdat het een nieuwe variant is op een bestaand model.

Nieuw in TPI Next

De volwassenheid matrix is aanzienlijk veranderd. Verdwenen zijn de soms wat verwarrend geplaatste letters A, B, C en D als aanduiding van het bereikte niveau; nu is iedere cel een controlepunt en wordt ook zichtbaar dat een bepaald niveau in de huidige situatie deels wordt bereikt. Ook is er tooling beschikbaar om de resultaten vast te leggen, met een aantal grafieken over de verhoudingen tussen de aandachtsgebieden. Ook een backwards compatibility tool is aanwezig om resultaten uit het verleden bruikbaar te houden. Een belangrijke verbetering van het model zijn ook de Enablers (bruggenbouwers naar de overige SDLC processen) en de Clusters (een verzameling van verbetermaatregelen die een stapsgewijze verbeteraanpak mogelijk maken). →

	Initial	Controlled				Efficient				Optimizing		
1 Stakeholder commitment	1	2	3	4	1	2	3	1	2	3		
2 Degree of involvement	1	2	3	4	1	2	3	1	2	3		
3 Test strategy	1	2	3	4	1	2	3	1	2	3		
4 Test organization	1	2	3	4	1	2	3	4	1	2	3	
5 Communication	1	2	3	4	1	2	3	1	2	3		
6 Reporting	1	2	3	4	1	2	3	1	2	3		
7 Test process management	1	2	3	4	1	2	3	1	2	3		
8 Estimating and planning	1	2	3	4	1	2	3	4	1	2	3	
9 Metrics	1	2	3	4	1	2	3	4	1	2	3	
10 Defect management	1	2	3	4	1	2	3	4	1	2	3	
11 Testware management	1	2	3	4	1	2	3	1	2	3		
12 Methodology practice	1	2	3	4	1	2	3	4	1	2	3	
13 Tester professionalism	1	2	3	4	1	2	3	4	1	2	3	
14 Test case design	1	2	3	4	1	2	3	4	1	2	3	
15 Test tools	1	2	3	4	1	2	3	4	1	2	3	
16 Test environment	1	2	3	4	1	2	3	4	1	2	3	

Volwassenheidsmatrix



De toekomst

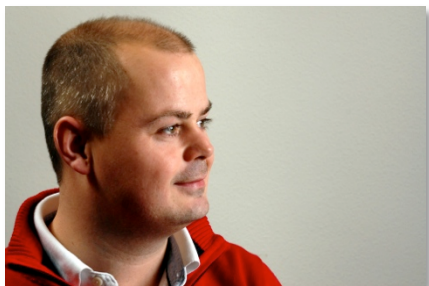
Momenteel wordt gewerkt aan TPI Next *Masters*. Dit is een programma dat voorziet in het nationaal en internationaal aanbieden van een aantal diensten gerelateerd aan TPI Next. Hierbij kan worden gedacht aan zaken als het certificeren van processen en organisaties door middel van een TPI assessment, het certificeren van TPI consultants en het opbouwen van een database met ervaringscijfers. Hiermee kan dan een organisatie zijn resultaten vergelijken met die van de branche of vergelijkbare groepen.

Hulpmiddelen

Naast de boeken zijn er ook tools beschikbaar om de resultaten van een assesment direct vast te leggen in een volwassenheidsmatrix. Dat en nog meer is te vinden op www.tpinext.com. Heb je vragen of wil je reageren: graag! Het adres is: tpinext@sogeti.nl. ←

TESTERS STAAN BOVEN DE WET

Door Menno Loggere • menno.loggere@cerios.nl



Velen van ons herinneren zich het rechtsgeschil tussen een werkgever en een werknemer over een snelheidsovertreding. Een werknemer van een koeriersdienst reed iets harder dan toegestaan. De snelheidsoverschrijding zou zijn veroorzaakt door onoplettendheid in combinatie met een te hoge werkdruk. De werknemer is door de kantonrechter in het gelijk gesteld, maar de Hoge Raad heeft de werkgever in het gelijk gesteld. De wet overtreden voor de baas? Stof tot discussie, maar wel zeer herkenbaar, zeker voor koeriers.

Helemaal niet bewust

Afgezien van het woon-werkverkeer hoeft een softwaretester (van junior tot testmanager) van nature niet veel te rijden voor het uitoefenen van zijn of haar vak. Overtreed jij als tester dan ook nooit de wet tijdens de uitoefening van je vak? Integendeel, een klein rondje langs de velden leert dat we massaal de wet overtreden. Waar een automobilist zich bewust is van een maximumsnelheid en we daar ook iedere honderd meter door hectometerpaaltjes aan herinnerd worden, zijn wij testers het ons vaak helemaal niet bewust dat we massaal de regels negeren.

De kans is groot dat jij vandaag de wet overtreedt! De Wet Bescherming Persoonsgegevens (WBP). De wet die ons als burger zoveel mogelijk beschermt tegen identiteitsfraude en waakt over onze privacy. Niemand wil dat iemand anders met zijn of haar creditcardgegevens online gaat winkelen. Gelukkig beschermt de WBP ons hiertegen en ziet het College Bescherming Persoonsgegevens (CBP) toe op naleving. Daar zijn we natuurlijk ontzettend blij mee, maar realiseren wij ons eigenlijk wel dat de wetten die ons tegen misbruik van persoonsgegevens beschermen bij het uitvoeren van ons testwerk massaal worden overtreden? Persoonsgegevens mogen namelijk alleen worden verwerkt wanneer dit 'strikt noodzakelijk' is en zo wordt testen vrijwel nooit door de wetgever gezien.

Top vijf

Toch hebben testers een heleboel excuses om zich niet te houden aan de WBP. De Testnet werkgroep Privacy en Testdata stelde een excuus top vijf samen, met excuses die jij misschien wel regelmatig gebruikt.

1. 'Ik heb een geheimhoudingsverklaring getekend'
2. 'Het is te complex om de testomgeving te voorzien van goede fictieve data'
3. 'Niemand maakt zich er druk om, waarom zou ik?'
4. 'Het is de verantwoordelijkheid van de opdrachtgever'
5. 'Een functioneel beheerder kan er toch ook bij?'

Zijn dit dan geldige excuses? Ben ik strafbaar? Om deze vraag te beantwoorden moeten we eerst iets meer weten over de verschillende gegevens die we onderscheiden en ook over de risico's die we lopen bij het onjuist gebruik van persoonsgegevens.

Natuurlijk persoon

In de WBP is gedefinieerd wat precies een persoonsgegeven is. Dit is elk gegeven betreffende een geïdentificeerde of identificeerbare natuurlijk persoon. Een boodschappenlijstje op zich is geen persoonsgegeven. Maar als dat boodschappenlijstje herleidbaar is tot het boodschappenlijstje van mevrouw De Bruin, is het ineens wel een persoonsgegeven geworden. →

Gezien de vele koppelingen in de huidige informatiesystemen is een willekeurig gegeven al snel een persoonsgegeven en daar ben je je vaak niet zo van bewust. De WBP maakt onderscheid tussen het verwerken van persoonsgegevens in het algemeen en het verwerken van bijzondere persoonsgegevens. Bij deze laatste categorie valt te denken aan godsdienst, levensovertuiging, ras, politieke gezindheid, seksuele geaardheid en strafrechtelijke persoonsgegevens.

Risicoklassen

Om de toepasbaarheid van de WBP te verduidelijken voor informatiesystemen is er door het CBP een nadere interpretatie van de WBP uitgewerkt. Deze uitwerking is in de volksmond bekend onder 'AV23': Achtergrondstudies en Verkenningen 23 van G.W. van Blarckom en drs. J.J. Borking.

In AV23 worden verschillende risicoklassen benoemd die bepalen hoe secuur er met de betreffende persoonsgegevens moet worden omgegaan. Het belangrijkste aspect dat de risicoklasse bepaalt, is de betekenis van de te verwerken persoonsgegevens binnen het maatschappelijk verkeer. Daarnaast spelen beveiliging van persoonsgegevens en privacybescherming een rol. Samen met de ICT-infrastructuur waarin de persoonsgegevens worden verwerkt, kan een persoonsgegeven zo in een risicoklasse worden geplaatst. Niet genoemd in de achtergrondstudie, maar wel iets om rekening mee te houden is dat ook een afbeelding (foto, video) een persoonsgegeven is. Een beeltenis is natuurlijk uniek, maar geeft ook informatie over bijvoorbeeld ras en geslacht. Zie het kader voor de risicoklassen.

Wat betekent dit dan voor testers?

Regelmatig lezen we in de krant dat er iets is misgegaan met grote gegevensverzamelingen. Betalingen bij een bank worden tweemaal afgeschreven, persoonsgegevens komen op straat te liggen, mensen krijgen onjuiste brieven, enzovoorts. Veel te vaak blijkt de oorzaak van deze problemen te liggen in het gebruik van productiedata bij het uitvoeren van testen in een testomgeving.

De verantwoordelijkheid voor het gebruik (of misbruik?) van persoonsgegevens ligt in eerste instantie bij de eigenaar van de gegevensverzameling. De persoonsgegevens worden aan haar verantwoordelijkheid toevertrouwd en zij dient er zorg voor te dragen dat de naleving van de WBP van verzamelen tot vernietigen is ingebed in de reguliere beheers- en verwerkingsprocedures. Dit vinden we terug in functiescheiding en bijvoorbeeld het protocolleren. In een productieomgeving wordt exact bijgehouden wie welke gegevens opvraagt. De functionaliteiten in productie- en testomgevingen zijn vrijwel gelijk, de inrichting blijkt echter vaak flink te verschillen. De functiescheiding in productie is er in de testomgeving vrijwel nooit. Testers hebben in de testomgevingen autorisatieniveaus die toegang tot alle gegevens geven. Ook wordt de beveiliging en protocollering vrijwel nooit ingericht zoals in productie. Toch wordt in alle omgevingen regelmatig productiedata aangetroffen. De verleiding om eens in de bankgegevens van je buurman te kijken is zo veel groter dan in een productieomgeving, waar je grote kans loopt betrap te worden.

Maak dus geen gebruik van persoonsgegevens die in de hoogste risicoklassen vallen. Het komt in de praktijk vaak voor dat we als testers alleen maar productiedata ter beschikking krijgen en dat een opdrachtgever van een testtraject geen budget heeft geclaimd voor het werken met goede geanonimiseerde testdata. Weigeren om je werk te doen is natuurlijk ook voor de carrière van een tester geen goede zaak. Wat kun je dan wel doen? →

Bepaal voor jezelf of jij en je organisatie kunt en wilt testen met fictieve data of met (een kopie van) productiedata. Als je test met productiedata, hoor je per definitie extra voorzichtig te zijn. Wees je bewust van de gevaren die het verlies van persoonsgegevens met zich meebrengt. Geen van de excuses uit de excuus top vijf zou je mogen gebruiken als reden om toch met persoonsgegevens te werken. Als tester dien je de opdrachtgever te wijzen op de gevaren van het testen met persoonsgegevens. Geef duidelijk aan wat de gevaren zijn van deze inrichting van je testomgeving. Het is mogelijk om de productiedata te anonimiseren, zodat de WBP hierop niet meer van toepassing is. Houd je aan je testactiviteiten en ga geen gegevens verzamelen waar je als persoon in geïnteresseerd bent. Zoek nooit bekenden op in de systemen, dit kan verregaande consequenties hebben.

Neem als tester je verantwoordelijkheid. We zullen nooit kunnen voorkomen dat er een persoonsgegeven ongewenst op straat ligt, maar we kunnen er zeker een belangrijke bijdrage aan leveren!

De werkgroep Privacy en Testdata van Testnet heeft tot doel om op een juiste manier om te gaan met testdata. Dit gebeurt in een aantal stappen. Allereerst zetten we het probleem op de kaart, we gaan bewustwording creëren. Vervolgens wordt een visie en gedragscode opgesteld. Afgesloten wordt met het samenstellen van een toolkit die ondersteuning biedt bij het maken van goede testdata. ←

Risicoklasse 0 Publiek niveau

Dit zijn gegevens waarvan 'algemeen aanvaard is dat deze, bij het beoogde gebruik, geen risico opleveren voor de betrokkene'. Dit zijn gegevens die voor iedereen beschikbaar zijn in bijvoorbeeld de Gouden Gids en op publieke internetsites.

Risicoklasse I Basis niveau

In deze klasse horen gegevens waarvan 'de risico's voor de betrokkene bij verlies of onbevoegd of onzorgvuldig gebruik van de persoonsgegevens zodanig zijn dat standaard (informatiebeveiliging)-maatregelen toereikend zijn'. Dit zegt ons nog niet zo heel veel. In de praktijk zijn dit gegevens die op basis van normale relaties gebruikelijk zijn vastgelegd. Voorbeelden zijn de ledenadministratie van een vereniging of school met de relatie vereniging – lid, school – leerling, of een winkel die gegevens van klanten voor facturatie vastlegt.

Uitzonderingen hierop zijn gegevens die vallen onder de categorie bijzondere gegevens. Bijzondere gegevens geven informatie over bijvoorbeeld politieke voorkeur, lidmaatschap van een vakbond en iemands geaardheid.

Risicoklasse II Verhoogd risico

Voor gegevens in deze risicoklasse bestaan er 'extra negatieve gevolgen voor de betrokkene bij verlies, onbehoorlijke of onzorgvuldige verwerking van de persoonsgegevens'. Enkele voorbeelden die door de achtergrondstudie worden genoemd:

- ras;
- geloofsovertuiging;
- gegevens over de economische situatie van de betrokkene;
- gegevens over kredietverstrekking;
- de impact van op zich onschuldige gegevens over een groot aantal betrokkenen (bijvoorbeeld de ledenadministratie van een vereniging met 100.000+ leden).

Risicoklasse III Hoog risico

Gegevens in de hoogste risicoklasse zijn 'gegevens die een dermate vergroot risico voor de betrokkene opleveren dat het gerechtvaardigd is deze verwerking in risicoklasse III te plaatsen'. In deze risicoklasse vallen onder meer gegevens waarop een bijzondere geheimhoudingsplicht van toepassing is. Denk hierbij aan medische gegevens of gegevens van opsporingsdiensten.

ER ZIJN MEERDERE IJSBERGEN OP DE WERELD

Door Eibert Dijkgraaf • eibert.dijkgraaf@capgemini.com



Vijftien jaar TestNet: wat moet er nu nog gebeuren?

Nadat testen zich heeft ontwikkeld als vakgebied en aantoonbaar flexibel mee te gaan in veranderingen, zie ik met het oog op de toekomst nog één belangrijke horde. De tester moet de verbetering zoeken in het gebruik van andere vakgebieden. Hoe kan een verandermodel, de Change Management Iceberg, worden gebruikt bij de invoering van Business Driven Test Management?

Meer dan goed testen

Een verbetering van het testproces vraagt om meer dan goed testen alleen. Veel testers heb ik leren kennen als trouwe vakbroeders. Betrokken bij het te testen product, strijdend voor kwalitatief goede producten en zoekend naar verbeteringen in het testvak. De drukbezochte TestNet-evenementen met een grote diversiteit aan testonderwerpen zijn daar een sprekend voorbeeld van. Toch zie ik in de praktijk dat de meeste winst niet alleen zit in het (nog) beter gaan testen, maar ook in de juiste randvoorwaarden. In het TPI Next – model wordt een aantal aangemerkt als 'enablers'. Bij het goed inrichten van andere processen en de juiste afstemming met het testproces wordt ook het testen efficiënter. Voorbeelden hiervan zijn professioneel projectmanagement en goed configuratiebeheer. Een ander voorbeeld van een 'enabler' is verandermanagement. Een testprocesverbetertraject stopt niet bij het verbeterplan met acties om het volgende niveau van volwassenheid te bereiken. Het eindresultaat wordt zeker ook bepaald door het veranderproces dat plaatsvindt bij de implementatie van de gewenste verbeteringen. Het begint meestal bij de kennis en vaardigheden van de betrokken medewerkers. Investerings in opleidingen en certificeringstrajecten vormen vaak een niet onbelangrijk onderdeel van het verbetertraject. Maar is er ook sprake van 'vruchtbare grond' voor de gewenste veranderingen? De betrokkenheid van het management, in verschillende geledingen binnen de organisatie, speelt dan een rol. En de bereidheid tot verandering bij de operationele laag en bij het management is van belang. Om hier effectief mee om te gaan in verandertrajecten moeten we ook kennis gebruiken die zich buiten onze testhandboeken bevindt.

Voorbeeld: invoering van BDTM

Stel we nemen de implementatie van Business Driven Test Management (BDTM) als gewenste verandering. Daarbij wordt de teststrategie gebaseerd op in kaart gebrachte risico's. De testrapportages ontsijgen het niveau van aantallen uitgevoerde testgevallen of opgeloste bevindingen. Een juiste toepassing van BDTM betekent dat de opdrachtgever in zijn eigen taal wordt verteld wat de risico's zijn. Dit is wel op basis van informatie uit een inzichtelijk testproces.

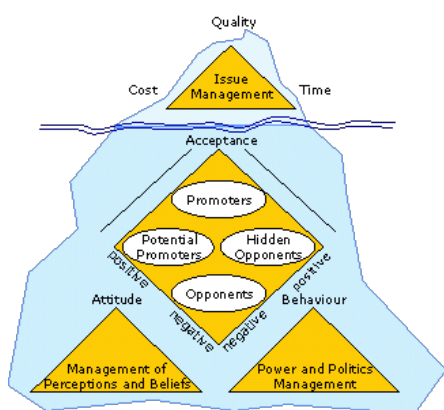
Bij de implementatie van BDTM komen vier testers op ons pad. Tester Piet Pro herkent de voordelen en wil het liefst zo snel mogelijk aan de slag. Zijn collega Antje Sjeens echter vindt het allemaal maar onzin. Ze doet al jaren haar werk, het systeem draait en als het fout gaat, wordt dat wel weer gecorrigeerd. Zij heeft ook nog een collega Bo Neus die er het zijne van denkt. Bo merkt wel dat het management veel waarde hecht aan een succesvolle implementatie. Dat is voor hem reden genoeg om zich te voegen naar de gewenste vorm. Ten slotte is er de broer van Bo, Bis Neus. →

Bis is ooit enthousiast het testvak ingerold en vindt het geweldig om een bijdrage te mogen leveren aan een goedlopend systeem dat de helft van alle Nederlanders is staat stelt om gemakkelijk en snel de juiste informatie op te vragen. Zijn eigen organisatie is daarmee ook succesvol en door zijn enthousiasme heeft hij zelfs zijn jongere broer Bo ook het testvak in gekregen. Nu zitten ze zelfs in hetzelfde testteam. De voorgestelde BDTM-aanpak is bij Bis nog niet echt geland. De enorme tijdsdruk van de huidige planning belemmert Bis om zich eens rustig te verdiepen in de nieuwe aanpak; laat staan om bepaalde activiteiten eens lekker op te schudden en anders aan te pakken. Maar Bis maakt zich ook zorgen. Hem is namelijk verteld dat hij straks keuzes moet maken. Dat hij niet alles meer mag testen. Thuis heeft hij zijn zorgen al eens met Bo besproken, maar die lacht zijn zorgen weg. Bis heeft het gevoel niet serieus te worden genomen, en dat terwijl hij zich altijd sterk maakt voor het belang van de klant en de eindgebruikers.

De Change Management Iceberg

In bovenstaand voorbeeld is inzicht in veranderkunde van toegevoegde waarde. We kiezen in deze situatie voor de Change Management Iceberg van W. Krüger (zie figuur 1). Voor testers bijzonder herkenbaar, want bijna iedereen is bekend met het ijsberg-plaatje uit TMap Next. De ijsberg visualiseert het zichtbare topje, terwijl zich onder de waterspiegel nog veel meer bevindt. In het testproces een metafoor voor alle testactiviteiten, die zich niet in de (zichtbare) uitvoeringsfase bevinden.

Change Management Iceberg



In een veranderproces zitten de factoren die houding en gedrag bepalen ook onder de oppervlakte, terwijl de dagelijkse werkzaamheden zichtbaar zijn als het topje van de ijsberg. Wat we zien en kunnen meten bevindt zich in de top. Of we bereid zijn om mee te gaan in de verandering en hoe structureel de verandering is, wordt bepaald door factoren die minder zichtbaar zijn. In het dagelijks werk is iedereen bezig met de waan van de dag. Telkens moeten daarbij keuzes gemaakt worden in de driehoek van tijd, geld en kwaliteit. Dit alles valt onder 'Issue Management'. Op dit operationele niveau worden ook de verbeteracties uitgevoerd. Er wordt gewerkt aan kennis en vaardigheden. Sjablonen worden aangereikt om de gewenste

verandering te ondersteunen. Dit alles zijn zichtbare en meetbare activiteiten en bevinden zich dus boven de waterspiegel. Maar hoe succesvol is de verandering? Wat gebeurt er nog meer? Wat bevindt zich onder het wateroppervlak? De betrokkenen worden geraakt en zijn in meer of mindere mate bereid om mee te gaan in de verandering. Denk aan Piet, Antje, Bo en Bis. Naast Issue Management zijn er daarom nog twee andere gebieden waarop gestuurd moet worden:

- Management van percepties en overtuigingen → bepalend voor de houding;
- Management van macht en politiek → bepalend voor het gedrag.

Deze aandachtsgebieden bevinden zich verder onder de oppervlakte. Houding en gedrag zijn daarbij wel zichtbaar, maar om deze te kunnen beïnvloeden is inzicht nodig in de achterliggende percepties en machtslijnen. Beïnvloeding is van belang omdat langs deze weg acceptatie van het gewenste gedrag tot stand kan komen. Dit maakt de vertaalslag mogelijk van een goed verbeterplan naar een succesvolle implementatie. →

De vier categorieën bij houding & gedrag

De Change Management Iceberg onderkent vier categorieën met betrekking tot houding en gedrag, die aan de hand van het eerder genoemde voorbeeld van de BDTM implementatie verder wordt toegelicht.

De Promoters hebben een positieve generieke houding ten opzichte van de gewenste verandering. Zij zien het belang voor de organisatie en daarnaast denken ze ook persoonlijk positief over deze verandering. De promoters halen voordeel uit de verandering en zullen deze daarom steunen. Tester Piet Pro is het type dat in de groep van de promoters geplaatst kan worden. Hij heeft een hoge acceptatiegraad en kan ook ingezet worden om anderen in zijn omgeving te beïnvloeden en mee te nemen.

De Tegenstanders (Opponents) hebben een negatieve algemene houding ten opzichte van de verandering zoals gewenst door de organisatie, maar ook een negatief gedrag vanuit persoonlijk perspectief. Tester Antje Sjeens bevindt zich in deze categorie. Zij is moeilijk te beïnvloeden. Ondersteunende maatregelen om haar gedrag te veranderen leiden vaak niet tot structurele verandering. Een terugval in de oude werkwijze ligt op de loer. De belangrijkste weg om Antje mee te nemen in de verandering is aandacht te hebben voor haar percepties en overtuigingen. Hoe komt het dat ze de nieuwe werkwijze niet ziet zitten? Welke beren ziet ze op de weg? In het geval van BDTM komt hier een aardig aspect naar boven bij testers. Testers kenmerken zich door een hoge betrokkenheid. Dat kan betrekking hebben op het product, de organisatie, de kwaliteit en/of de eindgebruiker. Deze betrokkenheid is vaak een prima uitgangspunt om tezamen aan het werk te gaan met de verandering. Het gezamenlijk vinden van hogere doelen die de intrinsieke motivatie van de medewerkers raakt, is een krachtig instrument om ze toch mee te krijgen. Het business perspectief biedt hiertoe vaak mogelijkheden. Bij de tegenstanders zal in de praktijk een 80-20 regel nuttig zijn, omdat niet alle betrokkenen overtuigd raken van de gewenste veranderingen. Om een verandering toch effectief te laten zijn, is het raadzaam om ook aandacht te hebben voor de groepen van verborgen tegenstanders en potentiële promoters.

De Verborgene Tegenstanders (Hidden Opponents) hebben een negatieve generieke houding ten opzichte van de verandering, hoewel zij deze oppervlakkig bezien wel schijnen te steunen. Het gedrag openbaart zich vaak wel in de gewenste vorm, maar doordat de diepere overtuiging ontbreekt, is de verandering niet van structurele aard. Deze groep kenmerkt zich ook wel als opportunistisch en tester Bo Neus is in het voorbeeld een representant van deze groep. Hij laat zich leiden door korte termijn voordelen, zoals een goede beoordeling of een bonus wegens goed gedrag. Bij vermindering van managementaandacht zal bij hem terugval naar de oude werkwijze een reële optie zijn. Ook hier is Management van Percepties en Overtuigingen noodzakelijk om zijn houding te veranderen. Meer dan de echte tegenstanders zal deze groep wel vatbaar zijn voor informatie en argumenten vanuit de dagelijkse praktijk. Bo Neus is structureel te beïnvloeden als vanuit het operationele vlak aangetoond kan worden dat de nieuwe aanpak daadwerkelijk efficiënter is of niet leidt tot meer productieverstoringen. Daarnaast zijn de verborgen tegenstanders ook vatbaar voor argumenten vanuit het Business Driven perspectief. Als in een gesprek de hogere gezamenlijke doelen benoemd kunnen worden, zoals het belang voor de klant of voor de eigen organisatie, kan de tegenstand worden weggenomen. Zo zijn verborgen tegenstanders te beïnvloeden om ook een positieve bijdrage te leveren aan de verandering.

De Potentiële Promoters staan in het algemeen positief ten opzichte van verandering, maar om bepaalde redenen zijn zij (nog) niet gekomen tot het zelf veranderen. Macht en Politiek management is in dit geval de passende lijn om een verandering teweeg te brengen. →

Tester Bis Neus heeft een hoge mate van betrokkenheid bij de eindgebruikers en de kwaliteit van het systeem. Dit is een goede basis om de verandering tot een succes te maken, maar deze betrokkenheid leidt ook tot praktische bezwaren, waardoor er toch belemmeringen ontstaan. Er zijn bepaalde drukmiddelen nodig om deze belemmeringen te overwinnen of weg te nemen. Dit kan door een positieve stimulans te geven. In het geval van Bis zal dit eerder tot een structurele verandering leiden dan bij zijn broer Bo. Ook kan het management tijdelijk randvoorwaarden scheppen om de negatieve consequenties te overbruggen, bijvoorbeeld door tijdelijk meer capaciteit in te plannen.

Krijg inzicht in de ijsberg

Samenvattend blijkt dat het gebruik van het business perspectief bij de teststrategie een krachtig instrument is. Niet alleen om uiteindelijk de opdrachtgever in zijn eigen taal te informeren, maar zeker ook om testers zelf mee te laten bewegen in veranderingen die voortvloeien uit testprocesverbetertrajecten. Een ijsberg kan daarbij een gevaar vormen als we ons niet bewust zijn van wat er zich onder de waterspiegel bevindt. Meer inzicht in de totale ijsberg maakt ons werk effectiever en efficiënter. Laten we ons er niet als een Titanic op stuk varen, maar laten we openstaan voor deskundigen van gerelateerde vakgebieden die ons kunnen vertellen wat zich onder water bevindt. Dan zal de tester de komende vijftien jaar zijn slagkracht nog veel verder kunnen vergroten. ←

OPDRACHTGEVERSCHAP EN TESTEN

Door Olaf Agterbosch en Johan Zandhuis • oagterbosch@sysqa.nl; jzandhuis@sysqa.nl  @OillyAg; @johanzandhuis



Ook opdrachtgevers hebben taken, bevoegdheden en verantwoordelijkheden! Een vaak gehoorde kreet van opdrachtgevers rondom ICT-projecten: 'Van ICT-projecten heb ik geen verstand en dat wil ik graag zo houden'. Deze opdrachtgevers delegeren opdrachten bij voorkeur aan een 'specialist'. Is daarmee de kous af? Bepaald niet. Ten eerste hebben de genoemde specialisten, over het algemeen softwarehuizen, veel verstand van ICT. Businesskennis hebben ze vaak minder. Als de opdrachtgever de opdracht helder formuleert, is de vertaling naar business requirements vaak onvoldoende.



De gevolgen daarvan voor de verdere vertaling in gebruikers- en systeemrequirements en de uiteindelijke oplossing laten zich raden. Slechte requirements zijn vervolgens de oorzaak voor slechte testspecificaties en dito eindproducten. Vaak is de opdrachtgever, zonder dat hij zich dat realiseert, een faalfactor. Hij is onbewust onbekwaam en delegeert wat onmiskenbaar zijn eigen taak is.



Figuur 1. De kunst van delegeren.

Outsourcing dan maar?

Outsourcing van systeemontwikkelactiviteiten, waaronder testen, is doorgaans een strategische beslissing, bijvoorbeeld om zich op de kerntaken van het bedrijf te focussen. Er zijn genoeg bedrijven waarvoor deze activiteiten wel een kernactiviteit is, zoals softwarehuizen. Gevolg hiervan is dat er een nieuwe relatie ontstaat tussen de opdrachtgever en de leverancier.

Het is verleidelijk te denken dat de opdrachtgever nu helemaal geen aandacht meer hoeft te besteden aan de formulering van de klantvraag. Het vertalen daarvan in ICT-termen is toch uitbesteed?

Ik zie de krantenkoppen al voor me opdoemen: 'Organisatie X verspilt Y miljoen euro aan mislukt ICT-project', of een variant daar op. Zeker bij outsourcingsituaties, maar feitelijk altijd, moeten opdrachtgevers zich realiseren dat hun betrokkenheid bij ICT-projecten verder gaat dan de klantvraag over de spreekwoordelijke muur te slingeren. Zaken als Quality Assurance en acceptatie, bijvoorbeeld door middel van een acceptatietest, horen ook bij goed opdrachtgeverschap. Een aardige metafoor voor een afbakening van de taken van een opdrachtgeverschap is de tempel van opdrachtgeverschap. Hierin zijn alle verantwoordelijkheden van de opdrachtgever in de fundering, pijlers en dak neergezet. De ruimte ertussen is het speelveld van de leveranciers. →



Figuur 2. De tempel van succesvol opdrachtgeverschap.

De verantwoordelijkheden van de opdrachtgever gaan dus verder dan alleen de eerste pijler: opdrachtdefinitie en acquisitie. En zelfs dat gaat vaak fout. Hoe is dit ten goede te keren? Uiteindelijk is dit allemaal terug te voeren op, aan de ene kant, de natuurlijke belangenverschillen tussen opdrachtgever en leverancier en aan de andere kant de aansturing van de ICT: de governance. Opdrachtgeverschap bestaat in een minimale opvatting uit het opstellen van requirements, de leveranciersselectie, monitoring en acceptatie. Maar als een opdrachtgever het goed doet, is het meer dan dat, zoals we zagen in de metafoer van de tempel.

Goed opdrachtgeverschap

Goed opdrachtgeverschap is dus steeds belangrijker. Uitbesteding en regievoering daarop gebeuren niet voor niets steeds meer. In toenemende mate proberen bedrijven zowel de echte kwantitatieve als kwalitatieve voordelen uit uitbesteding te behalen.

- Vergroting van business-ICT alignment. Een regieorganisatie kan de spreekwoordelijke Haarlemmer olie zijn voor de relatie tussen ICT-dienstverlening en de business. Een regieorganisatie heeft als kerntaak deze twee onderdelen op elkaar af te stemmen en moeilijk te verenigen belangen te overbruggen. De vraag aan de externe partij(en) wordt daadwerkelijk afgestemd op de behoefte van de business.
- In plaats van zelf te voorzien in de beschikbaarheid van voldoende gekwalificeerd personeel, kan door uitbesteding de zorg, met alles wat daar bij hoort, voor deze gekwalificeerde medewerkers verlegd worden naar een dienstverlener.
- Lager continuïteitsrisico. Een goed ingericht regiebureau of een goed ingerichte regieorganisatie draagt ertoe bij dat de externe dienstverlening zonder onderbrekingen kan plaatsvinden.
- Het streven naar een hoog kwaliteitsniveau kan worden gestimuleerd door een toeleverancier in te schakelen. In plaats van het eigen personeel opleidingen te bieden en vervolgens jaren te wachten om door de opgedane ervaring deze investering terug te verdienen, kan bij een dienstverlener de vereiste kwaliteit simpelweg worden gekocht, als tenminste de vereiste kwaliteit onderdeel is van de overeenkomst.
- Bestaande ingezette en toekomstig aan te schaffen bedrijfsmiddelen (machines, gebouwen, vervoersmiddelen, ICT-voorzieningen en dergelijke) kunnen worden afgestoten of juist niet worden aangekocht, wat extra, anders in te zetten kapitaal oplevert.
- Door een dienstverlener of toeleverancier in de arm te nemen vergroot men de flexibiliteit van de eigen organisatie, zodat die zich kan richten op haar kerntaken. Dit voorkomt het ontslaan van eigen personeel in moeilijke tijden. Bovendien vereist een ontslagronde relatief grote inspanningen en vaak veel geld, terwijl het beëindigen van een contract met een dienstverlener aan het einde van de overeengekomen contractperiode geen probleem oplevert.
- Regievoering zorgt ervoor dat de markt continu wordt bekeken, waardoor optimaal kan worden geprofiteerd van de markt. →

Testen

Testen is op zichzelf geen taak die valt onder een regieorganisatie, acceptatie wel. Het testproces is dus heel belangrijk voor de regieorganisatie die namens de opdrachtgever toeziet op de goede uitvoering van het testproces. Een goede opdrachtgever stelt graag zeker, dat de oplossingen die door ICT worden opgeleverd de juiste zijn. Dat betekent dat de testorganisatie belang moet hechten aan een goed inzicht in de verwachtingen van de opdrachtgever. Die verwachtingen worden onder andere geformuleerd in de acceptatiecriteria: welke concrete criteria gaan de belanghebbenden gebruiken om het projectresultaat te accepteren. Er kan veel verschil zijn tussen 'technisch werkend' en 'door de organisatie in gebruik genomen'.

Een lijst van acceptatiecriteria voor nieuwe ICT is idealiter vooraf geformuleerd vanuit alle belanghebbenden. Denk aan de beheerafdeling, commerciële afdeling, uitvoerende afdeling (gebruikers), financiële verantwoording (boekhouding), enzovoorts. Elke belanghebbende stelt acceptatiecriteria op waaraan de oplossing moet voldoen wil de belanghebbende de oplossing accepteren. Dit is een instrument in het kader van verwachtingsmanagement: door het uitspreken van voorwaarden wordt helderheid gecreëerd. Een praktisch punt is dat deze acceptatiecriteria in het stadium van testen simpelweg een voorwaarde zijn. Testen kan alleen goed gebeuren als je weet waar het product aan moet voldoen. Goede invulling van de opdrachtgeverrol is daarbij dus essentieel.

Bronnen

Agterbosch, O. et al. (2011). *Succes met opdrachtgeverschap!* SDU / Academic Service

Molen, M. van der (2009). *Prince2 voor opdrachtgevers*. Van Haren Publishing



HET IS NU TIJD OM DE BLIK NAAR BUITEN TE RICHTEN

Door Marcel Kwakernaak • marcel.kwakernaak@gmail.com



'TestNet streeft de professionalisering na van het testen van IT-producten.' Ondanks alle positieve ontwikkelingen in de laatste vijftien jaar kunnen we nog niet stellen dat we volwassen zijn. IT-projecten hebben een slechte reputatie als het om de kans van slagen gaat. Ook blijkt dat de top tien faalfactoren nagenoeg constant is gebleven. Hoe is het mogelijk dat we geen vooruitgang hebben geboekt? We zijn toch professioneel geworden. Kunnen wij het ook niet helpen, of zijn we wellicht kortzichtig? De kern van het probleem is dat veel IT-projecten te veel kosten en te weinig opleveren. De meest concrete bijdrage van testen voor de opdrachtgever is een goed onderbouwd 'Go/NoGo'- advies.

Dit is een 'verzekeringcertificaat' dat een extra waarborg is voor de business case van het project. Blijkbaar niet afdoende, want een positief vrijgaveadvies is nog geen garantie voor succes.

Initiatief nemen

Het gaat er niet om of testers verantwoordelijk zijn voor falende IT-projecten. Dat zijn ze niet. Het gaat er om wie in de aangewezen positie is om initiatief te nemen en een volgende stap te maken. Een stap in de richting van betere samenwerking en meer volwassenheid. Als we het vinden van fouten kunnen omzetten in het voorkomen ervan, zal dat zowel IT-projecten als het testvak ten goede komen. Samenwerking is de sleutel, hoe eerder hoe beter.

Samenwerken vereist een externe focus. Hoe kunnen we anderen helpen om ons te helpen? Hoe beter we de belangen voor de eindgebruiker begrijpen, des te beter we kunnen sturen. Hoe beter we laten zien wat we testen, des te beter samenwerkende partijen daarop kunnen sturen. Dat vraagt wel om een andere houding en een andere focus. De houding moet veranderen van reactief naar proactief. De focus van binnen naar buiten gericht, en naar de initiatie van projecten. Meer testen is niet de oplossing, meer samenwerken wel. Dat bevordert het wederzijds begrip, de rolverdeling, het vertrouwen, de toewijding en de kans op succes.

Veel projecten falen omdat er bij aanvang ernstige fouten worden gemaakt, die later niet meer kunnen worden hersteld. Dit betoog is gebaseerd op 'lessons learned'-evaluaties van IT-projecten. Vanwege onze centrale en overbruggende rol in het IT-speelveld zijn juist wij testers in staat om deze kar te trekken. →



Figuur 1. Schematische weergave van verleden, heden en toekomst van testen.

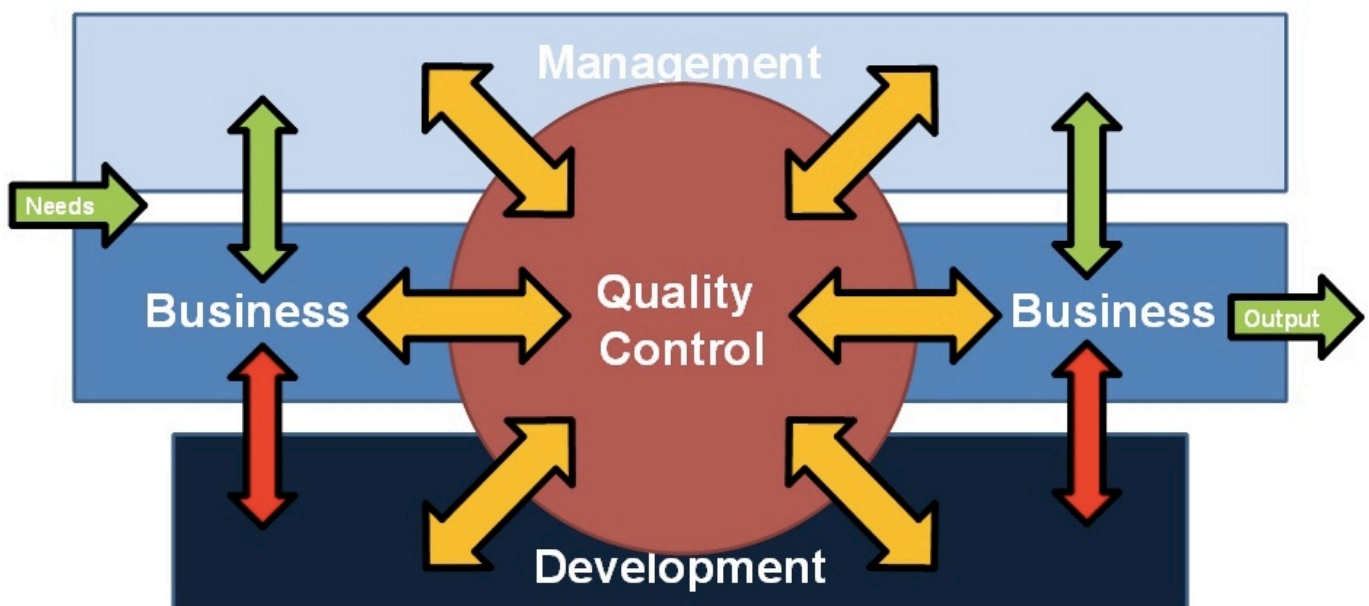
Wat betekent dit voor testprofessionals?

De blik naar buiten richten betekent dat we de aandacht verruimen en de prioriteiten verschuiven. Niet alleen interne verbeteringen doorvoeren, maar vooral de externe afstemming bevorderen. Veel testers doen dit al dagelijks, maar onvoldoende gestructureerd, consequent en herhaalbaar. Bekend obstakel is dat we in veel gevallen te laat betrokken raken. Dit is een een bekend aandachtspunt dat door veel testers wordt herkend. Bij de start worden de doelstellingen, prioriteiten en spelregels vastgesteld. Belangrijk is dat we leren beargumenteren dat, hoe we eerder we meedenken, des te beter dat is om valkuilen te vermijden. Dit is het basisniveau van samenwerking dat we moeten nastreven, de samenwerking met het management, de sponsor van het project.

Het tweede niveau van samenwerking is met de business, de eindgebruiker van het project. Hoe duidelijker is wat zij nastreven en met welke prioriteit, des te groter de kans op succes. Het halen van de business doelen is synoniem aan project succes. Vroegtijdig, actief en continu eindgebruikers bij het project betrekken heeft direct een uitwerking op het vertrouwen dat de business heeft in de goede afloop. Dit is een kritische factor voor IT-projecten. Als de business betrokken wordt bij het project zal zij beter meewerken bij de ingebruikname.

Het derde niveau van samenwerking is met development, de architecten en ontwikkelaars van het project. Met deze samenwerking zijn veel testers vertrouwd. Wat hier minder vanzelfsprekend is, is de vierkantscontrole: Het continu afstemmen en integreren van requirements, acceptatiecriteria, functionele en technische ontwerpen met systeem(integratie)- en acceptatietesten. Het doel van deze samenwerking is te verzekeren dat we op dezelfde doelen afkoersen en niet later in het traject allerlei inzichtverschillen hoeven op te lossen.

Wat vaak prima verloopt, is de interne teamsamenwerking. Rollen en taken worden verdeeld en waar nodig helpt men elkaar. Wat soms achterwege blijft is aandacht voor de externe samenwerking tussen de teams. Extra complicerend is dat projecten tijdelijke organisaties zijn, bij aanvang niet op elkaar ingespeeld. Vaak is er behoefte aan snelheid, haast om uit de startblokken te komen. 'Haastige spoed is zelden goed' klinkt spruitjesachtig, maar is wel van toepassing. Wel zijn urgentiegevoel en processnelheid positieve factoren. →



Figuur 2. Samenwerking: Groen gaat goed. Rood kan lastig zijn. Oranje is de oplossing.

Hoe kan dit werken?

Samenwerken begint met communicatie. Met behulp van communicatie kunnen de gemeenschappelijke doelen worden vastgesteld. Vervolgens moeten de rollen worden verdeeld en de bijbehorende verantwoordelijkheden worden genomen. Doorslaggevend is een gedegen voorbereiding. Dit maakt een goede start en vroegtijdig succes mogelijk. Het is het beste recept voor vertrouwen en een goede voedingsbodem voor vervolgsuccessen. Tot slot is ook de open mind en creatief leiderschap onmisbaar. Niet het plan maar het doel is heilig. Dit is een bondige omschrijving van de '6C Critical Success Factors': Communication, Common Objectives, Collaboration, Commitment, Continuous Quality and Creative Solutions. (Het 6C CSF model is in ontwikkeling. Input en commentaar zijn welkom.)

Samenwerken doe je niet alleen, het vereist een ontvankelijke omgeving en een open cultuur. Aanvankelijk zal in een gelimiteerd aantal gevallen bij aanvang van projecten de samenwerking worden voorbereid en afgestemd. Laat dat ons er vooral niet van weerhouden om gewoon te beginnen. Dit is een oproep voor actie! Test en quality professionals zijn de aangewezen kandidaten om dit initiatief nu te nemen en de aandacht te verleggen van professionalisering naar professionele samenwerking. Het vergt wel dat we ons aanpassen en de oriëntatie veranderen. Naar binnen gericht omdraaien in naar buiten gericht. Achteraf controleren naar vooraf adviseren, van testprocesverbetering naar IT-projectverbetering en van volgend naar initiatief nemend. De komende vijftien jaar liggen open en zijn voor ons ... als we dat willen!

En wat nu?

Testers werken al samen met development en zoeken al input van de business. Ze zijn op zoek naar de valkuilen die in het product besloten liggen. Dat hebben ze nodig om hun werk te kunnen doen. Als er meer structuur en stelselmatiger aandacht aan wordt gegeven, dan kan het alleen maar beter gaan. Hier ligt een dubbele kans. We moeten het ruimer zien, de blik naar buiten richten. De eerste kans is dat het niet om de kwaliteit van het testen gaat maar om de kwaliteit van het gehele project. Zo kunnen we ons aandachtsveld verbreden. De tweede kans is het breder toepassen van de methode die we zelf hebben ontwikkeld. Die van vragen stellen en terugkoppeling geven. Niet alleen om het testproces te sturen maar evengoed alle andere processtappen. Samenwerking is tweeweg verkeer. Niet alleen support vragen maar het juist ook het aanbieden.

Praktische tip: Organiseer workshopsessies gericht op kennisdeling. Ook hierin zijn we ervaren: een goed voorbereide en uitgevoerde PRA-sessie is een eyeopener voor alle deelnemers. Ditzelfde basisprincipe kunnen we zonder meer voor projectrisico's toepassen, zowel op strategisch als op praktisch niveau. Laten wij het voordeel van onze centrale positie en ons vermogen om een gemeenschappelijke taal te spreken benutten en de brug slaan. Wees creatief en neem initiatief! ←

TESTERS, DE MODERNE PROEFMEESTERS

Door Asad van Gelderen • Asad.VanGelderen@sqs.com  @SQSNL



Testers kunnen worden gezien als de moderne proefmeesters, maar helaas nog niet helemaal. Testen vindt al eeuwenlang plaats. Gedurende honderden jaren heeft de tester, soms onder verschillende namen, zijn werk uitgevoerd.

Gilde

Even een kleine stap terug in de geschiedenis. Al in de veertiende eeuw kenden de verschillende gildes meesterproeven.

Voor de goudgilde bijvoorbeeld was de minste kwaliteit goud die verwerkt moest worden, gesteld op achttien karaat (de zogeheten gout toets). Om dit te controleren moest de edelsmid zijn werk een merk geven (het meester-teken). Maar dat was nog niet voldoende. Er waren te veel edelsmeden die sjoemeld en andere (goedkopere) metalen mee smolten. Daarom moest het werk van de edelsmid door een derde, onafhankelijke persoon worden gecontroleerd. Dit was het werk van de proefmeester, de persoon die bij een gilde zorgde voor de handhaving van de kwaliteit. De proefmeester moest het werk, na goedkeuring, ook een teken geven.

De proefmeester, lid van hetzelfde gilde (dus niet helemaal onafhankelijk), bekeek zaken zoals het toepassen van het juiste richtlijnen ten aanzien van materiaal en alle andere bepalingen die van toepassing waren op het werk van de gildeleden. Daarnaast was de proefmeester ook verantwoordelijk voor het afnemen van het toelatingsexamen van aspirant gildeleden (proefknechten).

Je werd niet zomaar proefmeester. Over het algemeen werd iemand met veel ervaring binnen het gilde proefmeester. Naast het hebben van ervaring moest de aspirant proefmeester de zogeheten meesterproef doorstaan. Proefmeester: een rol waarin iemand met een kritische blik, oog voor detail, communicatieve vaardigheden en kennis van producten en processen zeer goed tot zijn recht komt. Een rol voor iemand die deskundig is en passie heeft voor kwaliteit.

De mening van de proefmeester was belangrijk en een positief advies van hem was noodzakelijk om verder te mogen. De proefmeester in zijn rol als bewaker van de kwaliteit zal ongetwijfeld flink onder druk zijn gezet door de gildemeesters om te beslissen of een product al dan niet voldeed of dat een afgekeurd product wellicht toch verkocht zou kunnen worden (wellicht met een kleine aanpassing). Proefmeesters waren over het algemeen niet de meest geliefde personen. Voorbeelden zijn er ten over, bijvoorbeeld in de notulen van het gilde der chirurgijns te Amsterdam uit de jaren 1727 – 1760). Hun rol liet het niet toe om de populaire jongen uit te hangen. Ze waren immers verantwoordelijk voor de opgeleverde kwaliteit van hun gilde en dat namen ze zeer serieus.

Minimum kwaliteit

Ondanks dat ze niet de meest geliefde personen binnen het gilde waren, hebben de proefmeesters het uiteindelijk tot 1798 (ruim vierhonderd jaar!) volgehouden. Als de gildes in dat jaar niet waren opgeheven, waren er waarschijnlijk nog veel langer proefmeesters geweest. Waarom proefmeesters zo belangrijk waren, hoeven we jullie niet uit te leggen. De gildes hadden de proefmeesters nodig en omgekeerd. Zonder het leveren van een bepaalde minimum kwaliteit zouden de gildes niet hebben kunnen overleven. Dat hoort bij het volwassen worden van een proces en product. ➔

Hoe passen wij in dit plaatje? Wij testers zijn de moderne proefmeesters. We zijn positief kritisch, inhoudelijk sterk, communicatief vaardig en hebben een goed oog voor details. Testers hebben zeer zeker een passie voor hun vak en stralen dat ook uit. We beschikken over veel kennis van de processen en producten waar we aan werken en leveren een positieve bijdrage aan het systeem of product dat op de markt komt. Met het gebruik van de sociale media en andere moderne communicatiemiddelen wordt ons werk ook belangrijker. Denk maar eens aan de gevolgen (niet alleen financieel, maar ook qua reputatie) als bijvoorbeeld mobiele telefonie en internet een aantal dagen niet werken.

Iedereen

Helaas zijn we op alle punten nog niet zover gevorderd als dat onze voorgangers, de proefmeesters, dat waren. Een paar voorbeelden. Iedereen kan tester worden (we kennen geen echte meesterproef 'Testen'). Hoewel er grote stappen zijn gezet op het gebied van opleidingen, certificeringen (wereldwijd meer dan 200.000 ISTQB gecertificeerde personen) is er nog behoorlijk wat kaf aanwezig dat van de spreekwoordelijke koren kan worden gescheiden. Dit kunnen we realiseren door softwaretesten niet alleen maar met een paar vrijwillige extra modules op een willekeurige IT-opleiding te laten onderwijzen, of via een certificering die je erbij doet op het moment dat je 'iets' met testen gaat doen.

Naar onze mening kunnen de gildes van tegenwoordig (de 'producenten' van de verschillende testmethodieken) hierin een hele belangrijke rol spelen. Niet zozeer door hun eigen methodiek te verkondigen, maar door het beste van alles te gebruiken en dat toe te passen. ISTQB heeft daar een eerste solide basis voor gelegd. De nieuwe ISO29119-normering die op dit moment wordt ontwikkeld gaat daar nog een stap verder in.

Gedegen opleiding

Als testbranche zullen wij de lat hoger moeten leggen voor nieuwkomers en bestaande testers. Zorg dat ze een gedegen opleiding krijgen en dat de aspirant testers over de juiste set van hard en soft skills beschikken. Er zijn genoeg bedrijven waar iemand zonder enige ervaring binnen een jaar gegarandeerd testcoördinator is. Dat is mooi voor de tarieven, maar als het fout gaat, beschadigt dat de reputatie van onze hele branche.

Alhoewel testen belangrijker wordt en ook aan de mening van een tester een zwaarder gewicht gehangen wordt, is onze inschatting nog niet bindend. Verre van dat zelfs ... Mischa Maliepaard heeft in een recent artikel op <http://www.testnieuws.nl/> uitvoerig betoogd welke uitdagingen een gemiddeld testteam voor de kiezen krijgt ten tijde van het eindadvies. Het lijkt warempel of de tijd op dit punt heeft stilgestaan. Dat toont maar eens te meer aan hoe belangrijk het is dat wij als testers onafhankelijk kunnen opereren.

Daarin zullen wij moeten afwijken van onze voorgangers, de proefmeesters, en leren van hun fouten. Zij maakten deel uit van het gilde en moesten de kwaliteit van hun eigen gildeleden toetsen: de slager keurde dus zijn eigen vlees. Onafhankelijkheid moeten we afdwingen en dat doen we te weinig.



Daarom moeten we op dit punt de hand in eigen boezem steken en ons afvragen of de (statistische) onderbouwing van ons advies niet aan verbetering toe is. Bovendien: zouden we niet moeten streven naar het geven van een bindend advies? Dat betekent namelijk dat we ons niet meer achter een bepaalde vrijblijvendheid kunnen verschuilen. →

Naar onze mening zal dat de kwaliteit van ons eigen werk zeer zeker ten goede komen. Wij worden gedwongen om nog kritischer naar ons werk te kijken en zullen beter in staat zijn een kwalitatief goed onderbouwd en onafhankelijk, bindend, advies te leveren. Wij staan aan het begin van een tijdsperk waarin we onze invloed kunnen uitbreiden. Wij allen zijn proefmeesters van het eerste uur, klaar om onze kwaliteitssgenen door te geven aan de generaties na ons. Dit is een grote verantwoordelijkheid die op onze schouders ligt, die handschoenen moeten wij zeker oppakken. TestNet kan hierin een hele belangrijke rol spelen. Wij kunnen de testbranche mobiliseren, informeren, uitdagen en beter maken. Als wij daarin slagen, zal TestNet ook nog zeker vierhonderd jaar bestaan! ←

THE FUTURE OF A TESTER: BROADEN OR SPECIALIZE

Door Erik van Veenendaal • eve@improveqs.nl



Can we predict the future?

I was at the StarWest conference recently, running a tutorial on PRISMA [1], where I attended a keynote from James Whittaker. He predicted that in five years from now there would be no jobs any more for testers. He also stated that 'users do not care about product quality' being as controversial as ever ... It made me think of a course I ran some fifteen years ago, where the developers stated that testing would no longer be needed as soon as they would change their process using the new tool.

It was interesting to have keynote and follow-up speaker kiwi David Hayman opening his talk with the statement, 'thanks James, but I work in the real world ...'. My thoughts are more in line with David Hayman, yes things are changing but only slowly and if we look back twenty years many of the things we were doing back then are still valid today.

Column

Can we predict the future? Interesting enough I wrote a column on trends in software testing [2] in 2002. In preparing for this column I was reading the old column once more to find out how we perceived the future of software testing almost ten years ago. What was new in 2002?

- Conferences were filled with talks on risk-based testing and testing was trying to relate to the business in order to get them involved in making choices. Ten years later, I believe this is common practice although many organizations are still struggling to make risk-based testing practical and fit-for-use.
- Agile was relatively new in 2002 and would have an impact on testing. Techniques such as use case, exploratory and module testing became more important. Testers thought again about the independence of testing. Again, I believe this has happened, although we are still trying in many projects to find the right balance between structured testing and agile.
- Finally, I discussed test certification, especially ISTQB. It goes without saying that this has happened with today almost 200.000 testers being certified world-wide. Again, the transition is not yet completed and today we are still waiting for the third level of the scheme, the expert level, to be fully operational.

We can recognize trends, try to predict the future but day-to-day practices only changes very slowly due to many reasons. One of them being that a change means that people have to change their behavior and leave behind things that they feel comfortable with. As a result there is resistance and change management activities are needed to make things happen.

Changes take time

Can we predict the future? Let's remember that in the real-world changes take time and only go very slowly. I cannot see a revolution coming in the forthcoming years. Take a look at the testing book 'The art of software testing' by Glenford Myers that dates back from 1979 [3]. Many of the principles and techniques that he describes are still the foundation of today's testing and taught in many testing courses around the world. →

An interesting quote in this context 'Just because everything is different doesn't mean anything has changed.' Nevertheless what are the today's trends in software testing that the (traditional) tester needs to be aware of and respond to?

- *Knowledge and skills* will be a challenge in the near future for many testers. It is just not good enough anymore to understand testing and hold an ISTQB certificate. We will not anymore work in the save environment of our independent test team. We will work more closely together with business representatives and developers helping each other when needed. Work as a team trying to build a quality product. It is expected from testers to have domain knowledge, requirements engineering skills , development scripting skills, and strong soft skills especially on communication and negotiation.
- As products are becoming more and more complex many so-called non-functional testing issues will become extremely challenging. At the same time the business, users and customers do not want to compromise on quality. To be able to still test non-functional aspects such as security, interoperability, performance, reliability highly *specialist* testers will be needed. Even more than today these specialists will be full time test professionals with in-depth knowledge and skills in one non-functional testing area only.

As stated, I do not predict a revolution in testing. However, there are trends and we gradually change. You have the option to either broaden your knowledge and skills as a test professional or to become a test specialist in a certain (non-functional) area. In order for these changes to be successful, I would recommend testers to dare to be different and embrace change. After all 'If nothing ever changed, there would be no butterflies.'

References

- [1] E. van Veenendaal (2012), Practical Risk-Based Testing: The PRISMA Approach, Uitgeverij Tutein Nolthenius.
- [2] E. van Veenendaal (2002), New developments and Trends, in: TestNet newsletter, Issue 6, No. 4.
- [3] G.J. Myers [1979], The Art of Software Testing, Wiley-Interscience. ←

COMBINATIE VAN DE FUNCTIE REQUIREMENTS ENGINEER EN TESTER

Door Jan Jaap Cannegieter • jcannegieter@sysqa.nl



Ontwikkelingen

De afgelopen jaren hebben allerlei ontwikkelingen effect gehad op het testvak. Een toenemende integratie van systemen waardoor we aan ketentesten moesten gaan doen. Uitbesteding van testen waardoor ons werk ineens in India wordt gedaan en wij testregisseur zijn geworden. Agile ontwikkeling waardoor we ineens in een multidisciplinair team zitten in plaats van in ons veilige testteam. Een ontwikkeling van tien tot vijftien jaar geleden was dat de functie van ontwikkelaar en tester werd losgekoppeld.

Steeds meer organisaties gingen testen echt als een professie zien 'wat de programmeur er niet even bij doet'. Een andere ontwikkeling buiten het testen van tien tot vijftien jaar geleden was het loskoppelen van de functies requirements engineer en functioneel ontwerper. Net zoals bij bouwer versus tester is het besef ontstaan dat een requirements engineer iets fundamenteels anders doet dan een ontwerper. Mooi, dan hebben we dat helder en kunnen we over tot de orde van de dag, toch?

Grote besparingen

Nee dus, verandering is de enige constante dus de functies van requirements engineer en tester ook. De laatste jaren zien we (of in ieder geval ik) een andere ontwikkeling: het in één project combineren van de functies requirements engineer en tester. Er zijn al organisaties die de functie van requirements engineer en tester combineren en die organisaties gaan hier na een eerste pilot mee door. Waarom? Omdat in de praktijk blijkt dat grote besparingen te realiseren zijn in termen van tijd en geld. Daarnaast zijn andere voordelen te halen. Laat ik dat even toelichten aan de hand van een persoonlijk voorbeeld. Toen ik tester was, kwam ik erachter dat testtrajecten een bepaalde cyclus doorlopen. En die begint bij lezen, heel veel lezen. Van allerlei documenten waarvan de opsteller vond dat ze helder en eenduidig waren, maar die ik maar gedeeltelijk begreep en waar ik een heleboel vragen bij had. Maar je moet die documenten lezen om de kennis op te doen die je nodig hebt om de testgevallen op te stellen. Moet je nagaan hoeveel tijd je kan besparen door degene die dit document heeft opgesteld de testgevallen op te laten stellen! Die hoeft die documenten helemaal niet te lezen! Dus besparing in termen van bestede tijd en doorlooptijd.

Kennisbehoud

Een voorbeeld van een ander voordeel is kennisbehoud. Nog even terug naar het voorbeeld van mijzelf toen ik de requirementsdocumenten aan het bestuderen was om daarna de testgevallen op te stellen. Ik weet niet hoe gelukkig jullie hand is als tester, maar ik pakte (kreeg) altijd net die systeemdelen waarvan de requirements engineer al was uitgestroomd. 'Omdat zijn inzet toch niet meer nodig was, hij had zijn werk toch immers afgerond?' En mijn vragen dan? Door de requirements engineer te laten testen blijft die kennis in persoonlijke vorm aanwezig binnen het project. En naar mijn mening is kennis in persoonlijke vorm beter dan kennis in documentatievorm. →

Overlap

Zelf heb ik met zowel testen als met requirements engineering ervaring. En wat me opviel toen ik begon met requirementsanalyse was dat de technieken die testers en requirements engineers hanteren behoorlijk op elkaar lijken. Als tester heb ik vroeger regelmatig status-transitiediagrammen opgesteld om daar mijn testgevallen uit af te leiden. Nee, die staan inderdaad niet in TMapNext, maar zijn wel een heerlijke techniek om tot je logische testgevallen te komen. Jaren later was ik een UML Activity Diagram aan het maken, niet precies dezelfde techniek maar toch kreeg ik een déjà vu. Natuurlijk is requirements engineering en testen niet hetzelfde, maar toch overlappen een aantal activiteiten en technieken!

Documentatie

Zijn er dan geen verschillen tussen requirements analyse en testen? Jawel hoor, zeker wel. De belangrijkste heeft te maken met informatie. Als tester baseer jij je werk op documenten of andere informatie, dat geeft een bepaalde mate van veiligheid en zekerheid. Als requirements engineer zijn er veel minder gestructureerde bronnen, je hebt minder 'grond onder je voeten'. En daar moet je mee om kunnen gaan. Natuurlijk, we kennen als tester allemaal wel een project waar de eerste fasen van het project zodanig beroerd waren uitgevoerd dat er helemaal geen requirements of een functioneel ontwerp was. En ik heb mijn eerste requirements opgesteld als tester. Ja, de gebruiker vond het ook wat raar dat een tester als eerste bij hem langs kwam wat hij nu eigenlijk wilde. Moest de bouwer dat niet eerst weten? Niet alle testers kunnen met de situatie van weinig tot geen documentatie omgaan en hebben behoefte aan de zekerheid van goede uitgangsdokumentatie. Die testers zullen het lastig krijgen als ze in een combinatiefunctie requirementsanalist-tester komen.

Kans of bedreiging?

Ter afsluiting van dit artikel nog een overdenking: is het combineren een kans of een bedreiging? Als deze ontwikkeling doorzet, is het immers de vraag of de requirements engineer het testen erbij pakt of dat de tester de requirements engineering erbij pakt. Voor degene die niet de persoonlijke kenmerken hebben om requirements engineer te worden of degene die zich in hun ontwikkeling beperken tot testen (of zich helemaal niet ontwikkelen) is dit wellicht een bedreiging, voor anderen is het wellicht een kans. Eigenlijk vind ik de vraag of iets een kans of bedreiging is niet relevant. Persoonlijk denk ik dat je altijd naar ontwikkelingen in je omgeving moet kijken en daar waar nodig jij je kennis en werkwijze aan moet passen. En is dit een ontwikkeling die doorzet? Vinden we het

De requirements engineer – tester als linking pin



over pak hem beet tien jaar net zo gewoon om de functie van requirements engineer en tester te combineren zoals we het nu gewoon vinden om bouwen en testen niet te combineren? Het zou zomaar kunnen en als dat zo is, kun je er maar beter op voorbereid zijn. Tijdens mijn presentatie op het Testnet voorjaarsevenement ga ik dieper op dit ontwerp in. Ik benoem meer voordelen en ga dieper in op de gevolgen. En hoe werkt dit in watervaltrajecten en hoe in agile trajecten? En wat moet ik doen om me hier goed op voor te bereiden? ←

BESTUURSMEDEDELING

Door Michiel Vroon • voorzitter@testnet.org



Afgelopen maart hebben wij een geslaagde Algemene LedenVergadering gehad, waarin het bestuur verantwoording aflegde over het gevoerde beleid in het afgelopen jaar en de plannen bekendmaakte voor het lopende jaar. Belangrijk speerpunt voor dit jaar is de uitbreiding van het gebruik van moderne communicatiemiddelen.

Wij willen meer gaan doen met onze website TestNet.org en het bijvoorbeeld ook mogelijk maken thema-avonden te gaan 'uitzenden' via de website. Daarom is tijdens de ALV het voorstel gedaan en aangenomen om een extra bestuursfunctie te creëren voor de inrichting en het beheer van dergelijke technische middelen.

Vol enthousiasme heeft Marco van der Spek deze rol op zich genomen en is hij voortvarend ermee aan de slag gegaan. Daarbij kwamen nieuwe zaken aan de orde, die Marco het gevoel gaven dat zijn TestNet-werkzaamheden zouden moeten concurreren met de werkzaamheden en plannen voor zijn eigen website TestNieuws.nl. Marco heeft aangegeven dat het voor hem onmogelijk is om beide zaken naast elkaar uit te voeren. Daarom voelde hij zich genooddaakt zijn net opgepakte taak als bestuurslid van TestNet weer neer te leggen. Wij betreuren de ontstane situatie en de gang van zaken daaromheen. Als bestuur gaan wij snel op zoek naar een alternatief. ←

EVENEMENTEN EN THEMA-AVONDEN

Voorjaarsevenement 15 jaar TestNet

Woensdag 30 mei 2012 09.00-21.00 uur

Locatie: NBC, Blokhoeve 1, Nieuwegein

Met in de morgen: vergroot je kennis en vaardigheden in een van de tutorials/workshops op het TestNet voorjaarsevenement 2012!! In een 3 uur durende workshop/tutorial (met een half uur pauze) neemt een ervaren spreker/trainer/coach je mee in een specifiek onderwerp en samen met de andere deelnemers breid jij je kennis en ervaring uit. En het is nog leuk ook !! De tutorials zijn uiteraard gratis voor leden, maar voor Tutorials moet je je wel afzonderlijk opgeven.

Vanaf 13.00 uur: TestNet bestaat 15 jaar !!! Met onder meer de presentatie van het boek van de "jubileum boek commissie". Je kunt het exemplaar dat je hierbij ontvangt laten signeren... Het evenement bevat verder een grote keuze aan presentaties over ons mooie testvak met 's avonds de keynote van Geoff Thompson en Bob van de Burgt over hun ervaringen met het testvak en de testwereld. De dag wordt afgerond met de bekende TestNetwerkborrel. Zie <http://www.testnet.org/evenementen.html>. ←

TestNet Interactieve avond Jubileumboek

Donderdag 14 juni 2012 18:00 - 21:00 uur

Locatie: NBC, Blokhoeve 1, Nieuwegein. ←

TestNet Noord: "Meertalig testen"

Dinsdag 19 juni 2012 18:00 - 21:15 uur

Locatie Hajé Heerenveen, Schans 65, 8441AC Heerenveen. ←

TestNet thema-avond: "Meertalig testen"

Dinsdag 26 juni 2012 18:00 - 21:00 uur

Locatie: NBC, Blokhoeve 1, Nieuwegein. ←

TestNet Summer School 2012

Woensdag 11 juli 2012 09:00 - 21:00 uur

Locatie: NBC, Blokhoeve 1, Nieuwegein. ←

Evenementen & Thema-avonden

E-mail: evenementen@testnet.org

Bernd Beersma

Eddy van den Berg

Erik Hendriks

Huib Schoots

Ine Lutterman

Peter van Tulder

Rik Marselis

Willem van Strik