

TESTNET NIEUWS

het online en interactieve testmagazine



Voorjaarsspecial 2017
20 jaar TestNet

TESTNET NIEUWS

Mei 2017 • Jaargang 21 • Voorjaarsspecial

www.testnet.org secretaris@testnet.org



VAN DE REDACTIE

Door Rob van Steenbergem • redactie@testnet.org [@rvansteenbergen](https://twitter.com/rvansteenbergen)



1997. Junior tester bij Centric staat op mijn LinkedIn profiel. Toen was de testafdeling van Centric nog een eigen organisatie op zich, QualityHouse. Dus eigenlijk ben ik ook 20 geworden dit jaar! Dus kan ik mijn jubileum ook vieren. En terwijl ik dit schrijf, heb ik dat op een speciale manier gedaan. Het organiseren met TestNet van het Nederlands Kampioenschap Software Testen 2017. De competitie heeft plaatsgevonden op 1 mei en we beschouwen het nu al als een succes. 23 teams hadden zich ingeschreven en hebben drie uur lang getest op een stuk software. Het harde werk voor de jury begint nu om alle rapportages te reviewen en alle bugs te bekijken, zodat we op 15 mei een Nederlands Kampioen Softwaretesten 2017 hebben! Dat maakt het twintigjarig jubileum nog specialer. Wellicht jammer voor mij, want dan verlies ik mijn titel als NK kampioen softwaretesten, maar goed. Dat mag ook wel na vier jaar.

Volgend jaar organiseren we weer een kampioenschap, begin maar vast met trainen. Om je wat van de sfeer te laten proeven hebben we wat foto's in deze TNN magazine gepubliceerd. Ik zie je graag op 15 mei om de kampioenen flink in het zonnetje te zetten.

En nu nog even genieten van ons twintigjarig jubileum magazine natuurlijk. Te beginnen met het enthousiaste verhaal van Rik! ←



De PO en jury van het NK Softwaretesten. Patrick (PO), Huib, Gerard, Klaas Durk, Rob en Pascal

COLOFON

Redactie

Gerben de la Rambelje
Gilbert Smulders
John Kronenberg
Kees Blokland
Lisa Gelijns
Paul Beving
Rob van Steenbergem
redactie@testnet.org

Bestuur

Rik Marselis	Voorzitter
John de Goei	Penningmeester
Peter van Tulder	Evenementen & thema-avonden
Ruud Teunissen	Informatievoorziening & beheer
Bernd Beersma	Marktverkenning & werkgroepen
Harro Philip	Secretaris & ledenadministratie

In dit nummer

De prehistorie	4
Twintig jaar TestNet, alsof het gisteren begonnen is	6
De verleden, heden en toekomst van T-shaped testers	8
Twintig jaar TestNet: Terug-en vooruitkijken	11
Kennisdeling in testen en jongleren	14
Testen met zelflerende en zelf-explorerende testtools	16
Reacties oud-leden over twintig jaar TestNet	19
#Duurzaam	23
De heuristiek van testverbetering	26
Twintig jaar TestNet: Niet alles is anders	29



TESTNET NIEUWS

het online en interactieve testmagazine

TESTNET

[Nieuws.testnet.org](http://nieuws.testnet.org) – TestNetNieuws wekelijks online

De TestNetNieuws 'Weekly' verschijnt iedere week in de vorm van één artikel op de website. Surf eens naar TestNetNieuws op [http://nieuws.testnet.org!](http://nieuws.testnet.org)

VAN DE VOORZITTER

Door Rik Marselis • voorzitter@testnet.org



TestNet is jarig!! Dus... FEEST ;-)

Als een groep mensen samen een passie heeft, wat doen ze dan? Praten, informatie uitwisselen, hun passie verder uitdiepen en verspreiden. En als dat een wat vastere vorm krijgt, richten ze een club op. Twintig jaar geleden deden Martin, Ingrid, Erik en Jos precies dat. Hun passie was testen en ze namen het initiatief tot een test-club. Bij de eerste bestuursvergadering werd al trots een aantal van 140 leden gemeld. Dus het was gelijk een succes. Maar die 140 mensen konden vast niet bevroeden, dat we nu, twintig jaar later, op weg naar tweeduizend leden zijn. En dat er iedere keer zeshonderd bezoekers naar een TestNet evenement komen.

Twintig jaar terug vonden veel IT'ers (die toen nog DP-professionals werden genoemd, DP = data processing) testen geen apart vak. Er waren dan ook nog nauwelijks testopleidingen en de eerste certificeringen (van ISEB) kwamen pas een paar jaar later. Met het TMap boek was de professionalisering van het testvak in 1995 net begonnen en het was dan ook vast geen toeval dat twee van de schrijvers van dat boek ook initiatiefnemers van TestNet waren.

Ik ben ervan overtuigd dat de oprichting van TestNet flink heeft bijgedragen aan de snelle opkomst van het testvak in Nederland, waarbij trouwens het jaar-2000-probleem ook een flinke duit in het zakje deed. Kijk maar eens wat we sinds 1997 hebben zien ontstaan in Nederland: Diverse vervolgen op het eerste TMap boek, boeken over TestFrame en RRBT, TestGoal, SmarTEST, Testen 2.0, Testdata management, TPI, en nog veel meer. Daarnaast spelen TestNet-leden ook in internationale initiatieven zoals ISTQB, TMMi en IREB een grote rol.

Gelukkig blijft onze passie voortdurend in beweging. Door de DevOps stroming is er weer discussie of testers eigenlijk wel nodig zijn, developers kunnen toch ook wel testen (yeah, right... ;-)

Wat mij betreft blijven de doelstellingen van onze vereniging dan ook onverminderd actueel.

Maar het testvak alleen is niet wat TestNet tot zo'n succesvolle vereniging maakt. Het enthousiasme van de leden, die veel bijdragen als vrijwilligers, plus de gezelligheid tijdens de vele bijeenkomsten speelt uiteraard ook een belangrijke rol. Daarom is TestNet de op één na grootste vakvereniging in de IT.

Jij bent het vast met mij eens dat TestNet een geweldige vereniging is en dat we nog veel meer jaren samen van het testvak gaan genieten!!

Dus lees met veel plezier deze TestNet Nieuws en kom naar het jubileumfeest op 15 mei 2017. Luister naar de crème de la crème van de testwereld en geniet van de twee knallers van keynote sprekers. En wie dat zijn, blijft tot het laatste moment een verrassing :-)

Veel testplezier!! ←

DE PREHISTORIE

Door Martin Pol • martin.pol@polteq.com



Het is 1980: 'Testen is niet nodig, want de specificaties zijn 100% en zullen weinig veranderen. Alle ontwikkelaars zijn goed opgeleid en maken dus weinig fouten en daarenboven voeren ze ook unittests uit. Iedereen volgt een solide standaard, de populaire SDM-methode. Er is een goede planning en voldoende tijd. En bovenal, we 'doen' QA, wat inhoudt dat iedereen die niet volgens de regels werkt ervanlangs krijgt'.

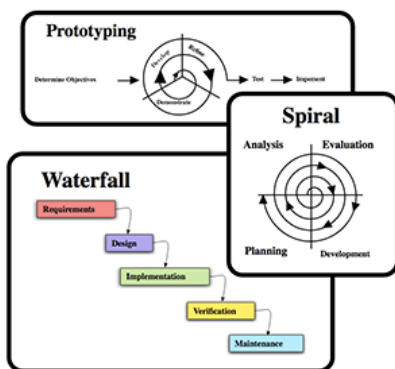
In sommige organisaties was men, onder druk van QA en EDP-auditing, meer bezig met het volgen van de regels dan met het daadwerkelijk ontwikkelen. Anderen deden maar wat. Er werd ongedisciplineerd gewerkt, prototypend en exploratory, in feite Agile. Maar toen bestonden Scrum of Kanban nog niet om er enigszins gestructureerd een succes van te maken.

Helaas bleek 'testen is niet nodig' een utopische gedachte. Daar kwam men door veel schade en schande achter. Projecten liepen zowel qua planning als budget uit de hand. Slechts een enkele deadline werd gehaald en de costs-of-repair en de personeelskosten overtroffen dikwijls het dubbele van de initiële begroting.



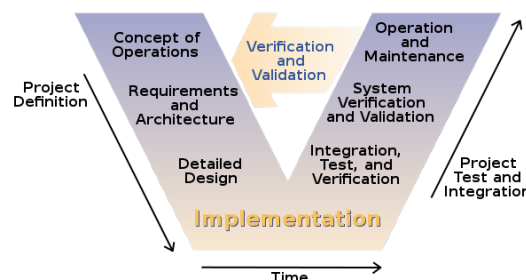
Testen toch wel nodig?

Omstreeks deze tijd begon men, vooral in de VS, met het ontwikkelen en toepassen van min of meer standaard methoden voor het testen, uitgevoerd door speciaal opgeleide testers die vaak in teamverband in projecten werden ingezet.



Dit zogeheten gestructureerde testen deed ook in Nederland haar intrede. Al watervallend probeerde men het gewenste product op te leveren en onafhankelijke testteams kwamen in zwang. Al snel werd het V-model het meest als template gebruikt voor het proces van ontwerpen, bouwen en testen. Dat was flink wennen. Voor ontwikkelaars was nieuw dat er iemand over de schouders meekiekte, die in veel gevallen ook nog eens hun werk afkeurde. Testers moesten, vaak als persona non grata, hun plekje in het proces veroveren. In eerste instantie alleen als acceptatietesters namens de gebruikersorganisatie, maar al snel ook meer als systeemtesters. De systeemtest schoot er bij de ontwikkelaars vaak bij in; daar was dus veel winst te behalen. Het verwerpelijke 'wij en zij' denken en handelen ontstond in alle hevigheid. Van het testteam werd verwacht een testplan inclusief betrouwbare planning op te leveren. Sommige testteams werkten zelfs fixed-price. Maar hoe doe je dat in een scene van stevast te late en incomplete opleve-

ring, voortdurend aangepaste specificaties en onvoldoende change control? Wat het testen vooral parten speelde, was de constante tijdsdruk, de heilige opleveringsdatum, terwijl de bijna onvoorspelbare hoeveelheid hertests voortdurend toenam. De projectleiding zag, zeker in aanvang, het testen als een vervelend noodzakelijk kwaad en had meestal weinig begrip voor de specifieke problemen van het testteam. Testen moest 'sneller'. Over 'sneller' programmeren werd niet gerept.



Testen is nodig

Vanaf de eerste structureringstappen heeft het minimaal een decennium geduurd voordat het testen een enigszins volwaardige plek in de systeemontwikkeling had veroverd. Standaards werden ontwikkeld en geïmplementeerd, opleidingen kwamen van de grond en de eerste testtools werden ingezet. Een project zonder testteam was nu ondenkbaar. 'Testen bleek nodig' en werd een vak.

TestNet

In dit tijdsframe ontstond het idee om als testers de handen ineen te slaan, breder dan binnen je testteam of eigen organisatie. Naar voorbeeld van de NGGO (Nederlands Genootschap voor Gestructureerd Ontwikkelen) werd het initiatief genomen de NGGT op te richten, met als doelstelling samen te leren, ervaringen uit te wisselen en vooral niet allemaal hetzelfde wiel uit te vinden. Tijdens de oprichting werd al snel een beter passende naam bedacht: TestNet! Het is 1997. ←



TWINTIG JAAR TESTNET, ALSOF HET GISTEREN BEGONNEN IS ...

Door Chris C. Schotanus • chris.schotanus@cgi.nl [@cschootanus](https://twitter.com/cschootanus)

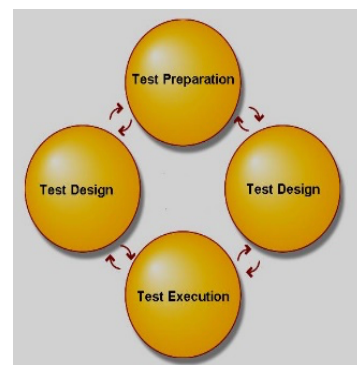


De twintig jaar dat TestNet bestaat, loopt grotendeels parallel met de ontwikkeling van volwassen testen in Nederland. We kunnen stellen dat de vereniging is ontstaan als gevolg van het volwassen worden van het testvak. Gedurende het grootste deel van die twintig jaar is er overigens niet heel veel veranderd tot voor een jaar of vijf, met de verdere acceptatie van Agile methoden. In dit artikel een persoonlijke blik op de historie.

De start van gestructureerd testen

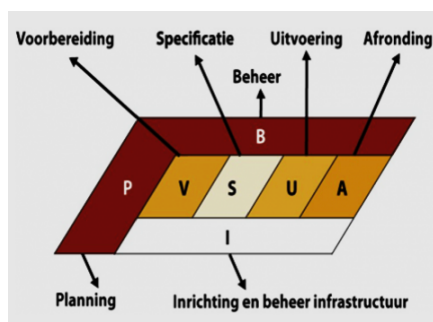
Halverwege de jaren negentig van de vorige eeuw zette ik mijn eerste schreden op het testvlak. Dat gebeurde overigens geheel toevallig; als medewerker van CMG (nu CGI) zat ik even zonder opdracht toen Hans Buwalda aan mij vroeg of ik wilde helpen bij een testproject. Mijn eerste reactie was: 'Testen? Dat doe je niet, dat laat je doen!'. Maar goed, bij gebrek aan een andere opdracht ging ik in het team van Hans aan de gang.

Hans had al de basis gelegd voor de methode voor testen van CMG: TestFrame. Tegelijkertijd kwam die andere grote testmethode op de markt: TMap van Iquip (nu Sogeti). Dat gebeurde beide rond 1995 en dat was volgens mij de start van het volwassen gestructureerd testen in Nederland en daarbuiten.



TestFrame Fasering

Methoden en technieken



TMap Fasering

Waar TMap zich uiteindelijk ontwikkelde tot dé standaard voor gestructureerd testen legde TestFrame de basis voor moderne technieken voor geautomatiseerde testuitvoering, keyword driven testing. Andere testmethoden die zich in Nederland en daarbuiten sindsdien ontwikkelden, zijn veelal gebaseerd op de structuur van TMap (ik laat ISTQB hierbij even buiten beschouwing; dat is mijns inziens geen testmethode maar een bundel gedegen standaards en technieken waarmee een methode kan worden gevormd). De moderne, al dan niet open source tools voor testen passen bijna alle het TestFrame principe van keyword driven testing toe.

De verdere ontwikkeling

Wat is er eigenlijk gebeurd sinds die beginjaren? Naar mijn mening eigenlijk maar weinig sensationeels. Er ontstonden inzichten voor het meer gebruiken van gevoel, ervaring en intuïtie binnen die zwaar gestructureerde methoden, denk aan exploratory testing en context driven testing. Ook gingen we zoeken naar technieken om het daadwerkelijke testen verder te automatiseren. Zo ontstond bijvoorbeeld model based testing.

De invloed van agile

De grootste veranderingen binnen het testproces zien we als gevolg van het invoeren van Agile en, in nog sterkere mate, DevOps ontwikkelen. Deze veranderingen raken, verbazend genoeg, bijna alle aspecten van het testen: testmanagement, de focus van de test, de toepassing van testtechnieken, de techniek bij de testuitvoering en de

rol van de tester. Ik pik hier twee aspecten uit: de focus van testen en de rol van de tester; volgens mij de belangrijkste veranderingen.

De focus van de test richt zich steeds meer op het zo vroeg mogelijk testen om zo snel mogelijk feedback te krijgen op de kwaliteit. Dat resulteert in veelvuldig componenttesten. Daarbij gebruiken we technische hulpmiddelen die rechtstreeks in de ontwikkelomgeving zijn opgenomen en die ons in staat stellen de softwarecomponenten volledig geautomatiseerd te testen. Bij DevOps, waarbij we uitgaan van continuous integration en delivery, is deze



Testing

© Mike Cohn, Mountain goat Software

Pyramid

aanpak een must; er mag bijvoorbeeld alleen correct werkende software in de repository worden ingecheckt. Aan de andere kant schuilt hier ook het gevaar dat er minder aandacht gaat naar het complexere testen op de functionaliteit doordat men van mening is dat, als de componenten goed werken, het gehele systeem ook wel goed zal werken. En uit ervaring kan ik vertellen dat dat niet zo is en dat de som van perfect werkende componenten niet per definitie een functioneel goed-werkend systeem oplevert. Google maar eens op 'testing pyramid', dan vind je Mike Cohn, die het model heeft beschreven en de vele reacties (positief en negatief) die daarop zijn geschreven.

En dan de verandering waar de tester het meeste van merkt: de rol van de tester. Deze verandert van kwaliteitspolitie in een teamlid dat vanaf het begin van de systeemontwikkeling bijdraagt aan het op juiste wijze invoeren van de systemen die op de juiste wijze de wens van de gebruiker vervullen. De tester krijgt minimaal een tweede rol; bijvoorbeeld als business analist, informatieanalist of als ontwikkelaar. Kortom, de rol van de tester verbreedt zich. En in die rol gebruikt de tester nieuwe technieken die zowel ondersteunen bij het omschrijven van de functionaliteit en testgevallen als mede bruikbaar zijn bij het geautomatiseerd uitvoeren van die testgevallen, technieken als Specification by Example.

Eindelijk: Shift left

Eigenlijk is wat we nu in testland zien gebeuren iets wat we twintig jaar geleden al heel graag wilden: we hebben de afgelopen twintig jaar steeds vaker ons best gedaan om als tester eerder betrokken te worden of betrokken te zijn bij het ontwikkelproces. Immers, Barry Boehm rekende het ons al lang geleden voor: zo vroeg mogelijk in het proces fouten vinden is relatief goedkoop. Dat we daar ons best voor moesten doen geeft aan dat de andere disciplines binnen het systeemontwikkelproces daar niet van overtuigd waren (zijn).

Wat we in ruim twintig jaar testen nauwelijks voor elkaar kregen blijkt door de invoering van Agile en DevOps eindelijk te gebeuren: de tester wordt betrokken bij het ontwerp, het vroeg in het proces opstellen van testgevallen en het continu testen van alle producten. Dit alles leidt tot 'first-time-right' leveren van systemen die daadwerkelijk doen wat de organisatie nodig heeft, wat willen we nog meer? ←

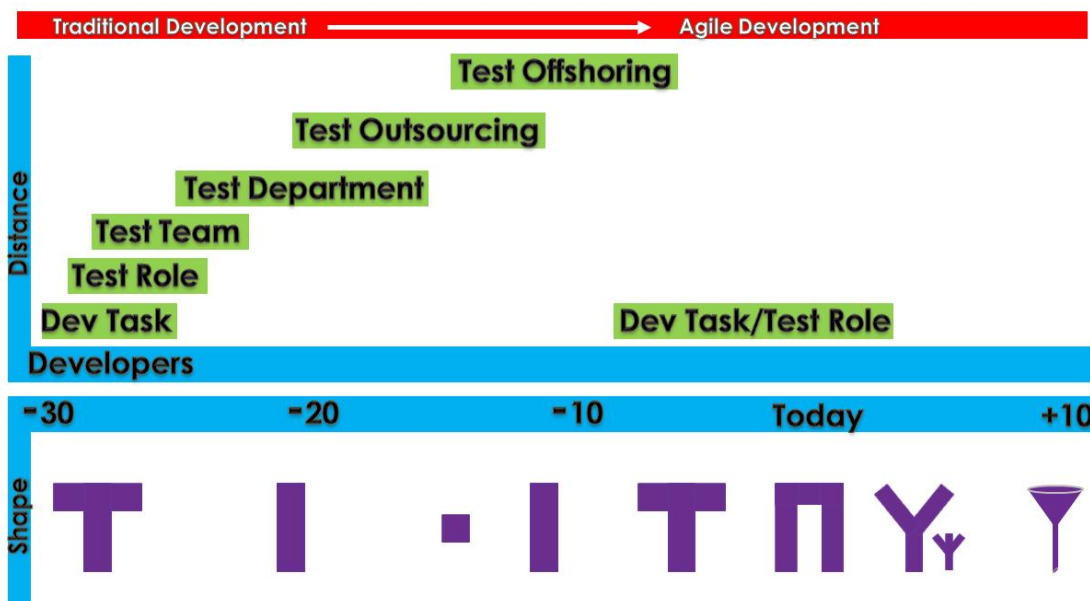
VERLEDEN, HEDEN EN TOEKOMST VAN T-SHAPED TESTERS

Door Leo van der Aalst • leo.vander.aalst@sogeti.nl



Vaak krijg ik vragen als: moet ik een I, T, n- or M-shaped professional worden om op de - nabije - toekomst voorbereid te zijn? Nou, ik denk het niet. Ik denk dat je eerder aan een Y-shaped professional, of beter nog, aan een antenna- of funnel-shaped professional moet denken! Maar voor ik dit verder toelicht, start ik met de IT-situatie van dertig jaar geleden en de steeds veranderende afstand tussen softwareontwikkelaars en testers tijdens die afgelopen dertig jaar. Dan extrapoleer ik dat - samen met de shapes - naar de toekomst.

T Dit is mijn persoonlijke verhaal en die van jou is waarschijnlijk anders dan die van mij. Het verhaal hangt af van het land waar je in woont, de organisatie(s) waarvoor je hebt gewerkt en ook van jezelf. Die van mij start dertig jaar geleden in Nederland (zie figuur 1). Ik was toen een programmeur. Maar ik deed meer dan dat. Ik maakte een functioneel ontwerp, een technisch ontwerp, ik programmeerde en testte de software helemaal alleen. Testen was een taak die door de ontwikkelaar werd uitgevoerd. Je zou kunnen zeggen dat ik T-shaped was. T-shaped in de zin van iemand die één expertise beheerst en kennis heeft van enkele andere expertises. En, vaak vergeten maar erg belangrijk, als je T-shaped wilt zijn: toon begrip voor mensen met een andere expertise dan die van jou (de term T-shaped is voor de eerste keer genoemd in 1991 door David Guest).



Figuur 1: Afstand tussen Developers en Testers

Een aantal jaren later werd testen een aparte rol. Vooral bedoeld om iemand, anders dan alleen de ontwikkelaar, een advies te laten geven over de kwaliteit van het product. Projecten gingen werken met testsoorten (unittest, systeemtest, gebruikersacceptatietest enzovoort) en er ontstonden testteams. Niet lang daarna begonnen organisaties na te denken over hoe je zo effectief en efficiënt mogelijk binnen allerlei verschillende projecten kunt testen. En zie wat er gebeurde, opeens schoten er testafdelingen als paddenstoelen uit de grond. In zo'n afdeling was ik een I-shaped professional. Ik was uitsluitend bezig met testen: dat was mijn primaire focus. Ik

sprak, behalve met testers, nauwelijks nog met andere projectleden. Een I-shaped persoon is iemand die één expertise uitstekend beheerst.

■ En daarna de volgende stap. Als alle testactiviteiten door een testafdeling worden uitgevoerd, wordt het makkelijk deze naar een ander omgeving (onder andere organisatie, land) te verplaatsen om bijvoorbeeld kostenreductie te realiseren. Dit proces wordt vaak outsourcing genoemd. De laatste fase - met betrekking tot de afstand tussen ontwikkelaar en tester - is offshoring. In zowel outsourcing als offshoring projecten zag ik testers degenereren tot dot-shaped professionals. Het was ze slechts toegestaan, óf testgevallen te ontwerpen, óf tests uit te voeren. Je kunt je voorstellen welke problemen dit met zich meebracht. Het duurde niet lang voordat de I-shaped tester weer zijn herintrede deed.

T Agile development werd 'hot' (onder andere met Scrum) en opeens veranderde alles. De afstand tussen ontwikkelaars en testers moest zo klein mogelijk zijn (dus geen outsourcing en zeker geen offshoring meer). Nog beter is het als de ontwikkelaars zelf gaan testen. Testen - samen met de T-shaped professional - was weer terug naar het startpunt, naar een taak of rol.

N Als ik kijk naar de huidige situatie, wordt er over n- en M-shaped professionals gesproken, waarbij een n-shaped professional een expert is in twee expertises en een M-shaped professional zelfs in drie expertises. Echter, hoe helpt dit - bijvoorbeeld een n-shaped professional - een project als een professional een excellent ontwerper en een excellent programmeur is? Deze professional kan geen twee dingen tegelijk doen, dus het project wordt niet versneld. Uiteraard wanneer deze professional aan het ontwerpen is en een programmeur wordt ziek of iets dergelijks, kan deze professional de programmeeractiviteiten overnemen (maar dan wel stoppen met ontwerpen). Persoonlijk denk dat het 'handig' kan zijn om een n-shaped of M-shaped professional te worden, omdat daarmee je kans op een baan wordt vergroot. Hoewel ik denk dat het niet meevalt om een echte expert in twee of meer expertises te worden.

Y Dus, als de I-, T-, n- en M-shaped tester niet de toekomst is, wat wordt het dan wel? Ik denk dat je minimaal een Y-shaped professional moet worden. Iemand die in staat is twee verschillende expertises te combineren en te gebruiken in één expertise. Stel, je bent een erg kundig tester. Als een Y-shaped professional moet je in staat zijn om - nieuwe - ontwikkelingen als BDD, ATDD, DevOps, IoT, robotica tot je te nemen, te leren, te gebruiken, en in en aan te passen in jouw rol als tester. Maar jouw expertise blijft onveranderd testen (of in algemene zin, softwarekwaliteit) en niets anders. Als ik dit naar de toekomst extrapolereer - in analogie met I-, n- en M-shapes (één, twee en drie expertises) - dan is Y-shape niet het eindstation, maar ontwikkel je door naar een antenna-shape of beter nog naar een funnel-shape. Met antenna-shape bedoel ik dat je in staat bent meer dan twee ontwikkelingen, expertises, enzovoort in jouw eigen expertise te integreren. Uiteindelijk moet je in staat zijn alles wat je ziet, hoort, leert en leest - indien zinvol natuurlijk - te gebruiken en te integreren in jouw eigen expertise. Ik noem dit funnel-shape

▽ Kortom, voorbij de I-, n- en M-shaped tester zie ik een toekomst voor de funnel-shaped tester. Een tester die is gespecialiseerd in softwarekwaliteit: een échte expert. Iemand die continu onderzoekt, nieuwe inzichten zoekt en nieuwe - niet per se IT-gerelateerde - ontwikkelingen volgt. Dit kan in feite alles zijn, zolang je maar in je achterhoofd houdt hoe dit jouw werk kan innoveren, hoe dit jouw expertise naar een nieuw - hoger - niveau kan brengen. En 'last but not least': Doe het! Bezoek als testexpert bijvoorbeeld testconferenties niet alleen om slechts te luisteren, maar zoek bij die conferenties naar aspecten waarmee je jouw eigen expertise kunt innoveren. Met andere woorden, 'Specialize, Explore, Innovate and Do it' (zie figuur 2).



Ik ben ervan overtuigd wanneer je een waar funnel-shaped persoon bent, zowel jouw collega's als werkgever jouw ongelooflijke toegevoegde waarde zullen (h)erkennen. Door simpelweg boven op nieuwe ontwikkelingen te zitten, nieuwe dingen te ontdekken en deze aan te passen of toe te voegen aan jouw eigen expertise, zal iedereen om je heen – ook het project – hiervan profiteren. Dit maakt jou een zeer populair teamlid en jouw expertiseniveau zal – net als je ster – continu stijgen. En 'last but not least', je zult altijd veel plezier houden in, en nooit verveeld zijn met, je eigen expertise, omdat je je continu verder ontwikkelt. ←

Figuur 2: Funnel-shaped professional



NK Softwaretesten 2017 - Overzicht van de zaal en teams die vragen stellen aan de PO

TWINTIG JAAR TESTNET – TERUG- EN VOORUITKIJKEN

Door Heini Veneberg • h.veneberg@cimsolutions.nl  @HVeneberg



Een jubileum is altijd een mooi moment om terug en vooruit te kijken. Twintig jaar TestNet en vijftwintig jaar CIMSOLUTIONS laten zien dat de testwereld steeds professioneler en volwassen is geworden. Deze volwassenheid zegt ook iets over de volwassenheid van ICT. Daarnaast kunnen we als testdiscipline groeien in de wijze waarop we testcases sneller kunnen realiseren en regie nemen: waar testen we wat.

Terugkijken

De ICT-ketens zijn de afgelopen jaren langer en daarmee kwetsbaarder geworden. Daarmee nemen de risico's tot lamleggen van Nederland toe. Oeps! Om die reden wil men voor de productiegang nog meer duidelijkheid over de kwaliteit, voorspelbaarheid en betrouwbaarheid van het systeem.

CIMSOLUTIONS kan inmiddels vanuit een vijftwintigjarig bestaan terugkijken op de ontwikkeling van de ICT. Als tester waren we vijftwintig jaar geleden heel druk met alles testen, met het complexer worden van de systemen was het onbetaalbaar geworden om alles te testen. De oplossing: risicoanalyses. Vanuit die basis werd bepaald waar de testcapaciteit op ingezet ging worden. Daarnaast was het lastig om in sommige organisaties duidelijk te maken wat de redenen waren om testers in te huren. Want met de redenatie, we kunnen beter enorm goede ontwikkelaars aantrekken, dan hoeven al die fouten ook niet gevonden te worden door testers. Deze redenatie is al regelmatig niet waar gebleken.

De uitdrukking: `Een tester brengt niets voort`, werd vooral veel gebruikt in organisaties die Line-of-code gestuurd waren.

Genoeg geklaagd!

Het heden

Als ik zelf terugkijk, dan zie ik vooral mooie, positieve veranderingen in de ICT. Deze veranderingen stellen ons nu in staat om te werken aan complexe ICT-oplossingen.

Organisaties zijn gaan investeren in Agile methodes en tooling. Het Agile werken heeft geleid tot een wendbaardere organisatie, mits het goed wordt toegepast. Het Agile werken legt meer verantwoordelijkheden bij het team. Nu zitten daar twee uitdagingen in:

- a. Organisaties moeten het Agile principe breder invoeren dan alleen in het Agile team, bijvoorbeeld door implementatie van SAFe;
- b. Agile geeft teams meer verantwoordelijkheid. Organisaties vragen zich op dit moment vaak af: Wat is deze commitment nu echt waard?

Naast het Agile werken zien we veranderingen in het samenvoegen van ontwikkeling en operations. Rollen veranderen, maar het type werk blijft.

Door de grote en complexe ketens moeten we ook sneller wijzigingen in productie kunnen brengen. Door middel van het begrip Continuous Delivery komen er allemaal tools beschikbaar om dat te realiseren.

Deze veranderingen bieden ook positieve veranderingen voor ons als testers. Jarenlang hebben we signalen afgegeven dat:

- a. Het veel tijd kost voordat er een nieuwe Build beschikbaar kwam;
- b. Testautomatisering belangrijk was.

Wat zien we nu: organisaties omarmen Continuous Delivery en schreeuwen om testautomatisering.

Vooruitkijken

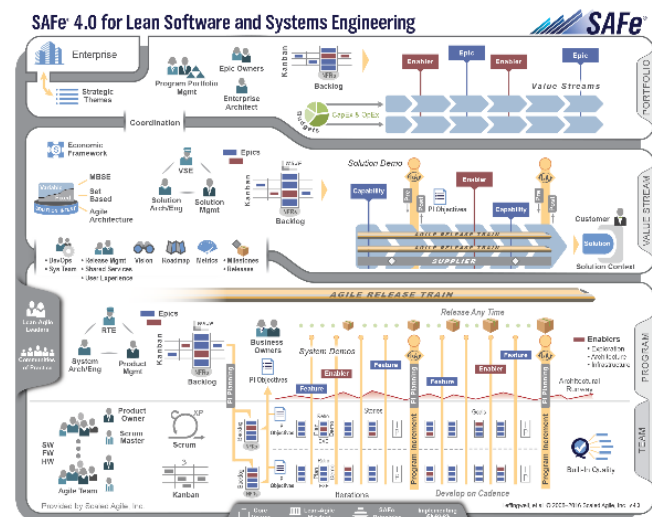
Het is toch mooi dat organisaties onze wensen van vroeger nu vervullen. Maar ben jij of zijn wij als tester nu klaar om goed met deze veranderingen om te gaan? Uiteindelijk moeten we er nu voor zorgen dat onze testcases sneller ontwikkeld kunnen worden, omgezet en toegevoegd worden aan de automatische testset. Ik denk dat daar onder andere onze uitdaging ligt voor de komende jaren.

Maar ook in:

- a. Regie nemen in waar testen we wat;
- b. Hoe zorgen we dat we qua testcaseontwikkeling niet op het kritische pad komen?

Regie nemen?

Als we in een SAFe organisatie een systeem ontwikkelen, kunnen we de regie nemen met betrekking tot wat we wanneer testen. En vooral: hoe zorgen we ervoor dat we bij de verschillende integratiemomenten het testresultaat van eerdere testen kunnen hergebruiken en niet alles opnieuw gaan testen. Met andere woorden, dat we afspraken maken met betrekking tot unittesten, componenttesten, moduletesten, producttesten tot en met het uiteindelijke systeem. Door middel van regie ga je vervolgens ook toezien op, en waar nodig corrigeren, dat het samengestelde testresultaat ook voldoende vertrouwen geeft zodat een goed en betrouwbaar systeem in productie kan gaan.



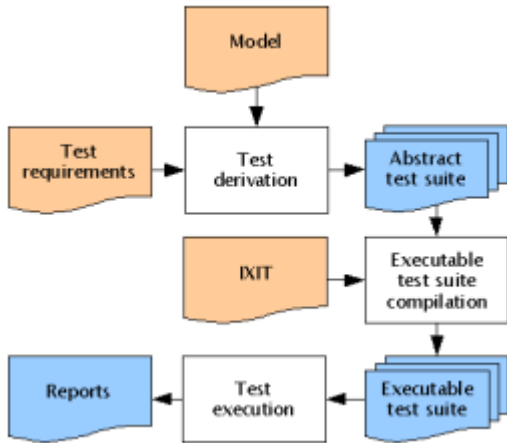
Hoe zorgen we dat testcaseontwikkeling niet op het kritische pad komt?

Door de veranderingen van de afgelopen twintig jaar zien we dat de toolset van de ontwikkelaars aan het veranderen is en daarnaast is er door middel van Frameworks al veel code beschikbaar, waardoor het wiel niet opnieuw uitgevonden hoeft te worden. Je ziet ook meer en meer modelleertechnieken. Het levert op veel plekken al geïntegreerde code op. Dit alles zorgt dat de ontwerp- en ontwikkeltijd korter gaat worden. Die tijd kunnen wij als tester volop gebruiken om testgevallen te bedenken. Nu deze tijd korter wordt, komt er een moment dat wij op het kritische pad komen. Dit in combinatie met handmatige testuitvoering komen we onder druk te staan in een Continuous Delivery tijdperk.

Een oplossing is al snel te geven: testautomatisering. Dit zorgt ervoor dat de testuitvoering vaker en sneller uitgevoerd kan worden. We zullen dus handig moeten worden in het omzetten van handmatige testgevallen naar toe-

voegen in de automatische testset. Waarbij ik wel wil vermelden dat de eerste testuitvoering handmatig cruciaal blijft, aangezien de mens slimmer en breder kijkt tijdens de uitvoering.

De andere oplossing zit in de testgevalontwikkeling.



Als testers moeten we gaan nadenken over hoe we aansluiten bij de ontwerpfase en middels Model Based Testing sneller testgevallen kunnen genereren. Nu bestaat MBT al even, maar de kunst is om dit op complexere onderdelen te gaan toepassen. Daarnaast kunnen we, door het gebruik van Frameworks, functionaliteiten op component- of serviceniveau testen en op systeemniveau voortborduren op het testresultaat. Dit zorgt ervoor dat we minder testgevallen hoeven te ontwikkelen. Dit alles kun je vanuit regiefunctie gaan inregelen.

Als laatste moeten we alert blijven op hoe organisaties Ownership van code inregelen. Op dit moment zie ik organisaties worstelen met feature gestuurd de Backlog vullen, in plaats van component gestuurd.

In het eerste geval zie je dat sommige teams vanuit feature gedachten zich niet verantwoordelijk voelen voor één of meerdere componenten.

Vanuit de positieve veranderingen van de afgelopen twintig jaar zie ik zeker nog volop werk voor de tester. Waarbij we Early adopter kunnen zijn en blijven qua testaanpak, echter moeten we ook heel bewust zijn dat we volger zijn afhankelijk van op welke wijze architectuur-, ontwerp- en implementatiewijze belangrijk zijn. Hoe moeten en kunnen we als tester hierop anticiperen en vooral mee veranderen. ←



NK Software testen 2017 - Een aantal teams hadden hun eigen t-shirt laten drukken

KENNISDELING IN TESTEN EN JONGLEREN

Door Han Toan Lim • info@mindfultester.com  @MindfulTester



Op 7 maart 2017 zei ik:

'I bought this magic trick at an international juggling festival in Pittsburg Pennsylvania.'
James Bach keek mij, een Nederlandse tester, belangstellend aan.

Hoe kwam het zo ver?

Na mijn studie had ik een baan gevonden met een reistijd van anderhalve uur enkele reis. Dat had wat gevolgen voor mijn vrije tijd. Wat kon ik doen in dat kleine uurtje tussen de afwas en het slapen gaan? De oplossing was jongleren boven mijn bed. Na mijn verhuizing ging ik trouw door met jongleren. Boven mijn bed.

Ik wou echte jongleerspullen en kwam terecht in een winkel in Amsterdam. Deze verkocht ook een magazine met informatie met jongleerbijeenkomsten. Ik was de koning te rijk. Ik werd lid van de Delftse jongleerclub. Binnen een paar jaar verbeterde mijn niveau van jongleren sterk. In 1995 ging ik naar Quality Week in London. Dit was de eerste keer dat ik testcursussen volgde. De onderwerpen waren testautomatisering en software inspecties volgens Gilb en Graham. Die week ging ik ook twee avonden naar Circus Space om te jongleren. London is the town for me.

Waar wachtte ik op?

Ergens in 1997 zag ik een mailtje dat een vereniging voor testen werd opgericht. Ik was net begonnen als tester en zette de vereniging uit mijn gedachten. Ik had nog kennis te absorberen bij mijn werkgever. In 2000 had ik diverse nationale en Europese jongleerfestivals achter de rug. Het was tijd om naar een Amerikaans jongleerfestival te gaan in Pittsburgh, Pennsylvania. Om kennis te delen. Een paar jaar later was het tijd om rond te kijken bij TestNet. Ik was nauwelijks binnen, toen ik een bekende zag. TestNet bleef me als tester trekken. Er werd gezocht naar een nieuwe voorzitter. Ik dacht aan mijn jongleerclubje in Delft. Ik ging helpen als bestuurslid bij TestNet.

Waar ging dat naar toe?

In de volgende jaren veranderden wat zaken: de Eenhoorn in Amersfoort werd ingeruild voor NBC in Nieuwegein. De leden- en financiële administratie werden uitbesteed. Het meest opmerkelijk was de groei van het aantal leden. Het waren vooral de leden, die TestNet groot maakten. Niet alleen door hun aantal, maar vooral door het delen van ervaring en kennis. Ze inspireerden mij om zelf op een paar TestNet bijeenkomsten te spreken. Het waren leerzame ervaringen, waar ik nu nog baat van heb. Dank jullie, TestNet leden.

O ja

Met een week te gaan voor de cursus van James Bach dacht ik terug aan Michael Bolton. Deze maakte ik voor het eerst mee op een TestNet evenement. James en Michael zetten mensen bij voorkeur op het verkeerde been.

In de notities van de gedownloade slides van de cursus, zag ik goochelen staan, dus ging ik in mijn dozen met oude trucs zoeken. Ik was



voorbereid.

Op 7 maart 2017 zei ik:

'I bought this magic trick at an international juggling festival in Pittsburg Pennsylvania.'

James Bach keek mij, een Nederlandse tester, belangstellend aan.

'That was my main reason to come', voegde ik hier aan toe.

James was even stil. ←



NK Softwaretesten - Team samenwerking op hoog niveau

TESTEN MET ZELFLERENDE EN ZELF-EXPLORERENDE TESTTOOLS

Door Rik Marselis (links) en Sander Mol rik@marselis.eu en sandermol@chello.nl



Op welke manier gaat machine intelligentie ons als testers in de toekomst verder helpen? Toen wij hier een tijdje over nadachten, kwamen we op het concept van zelflerende en zelf-explorerende testtools. Hieronder zetten wij deze gedachten uiteen. Aan het eind roepen we je op dit verder met ons te verkennen!



Testtools van nu, vooral voor het vaak saaie uitvoeringswerk

Testautomatisering automatiseert meestal de testcases die vooraf handmatig zijn opgesteld door de tester(s). Deze testgevallen hebben vooraf gedefinieerde resultaten. De tool verifieert of de werkelijke resultaten overeenkomen met de vooraf gedefinieerde resultaten. De testcases die worden geselecteerd, hebben als doel om een bepaalde dekking van alle mogelijkheden te behalen. Het aanpassen van het testobject betekent ook het aanpassen van de testcases in de tool. De testwereld staat nu op het punt om nieuwe, andersoortige automatische testtools toe te gaan passen. Tools die niet vooraf instructie hoeven te krijgen, maar zelfstandig een informatiesysteem verkennen.

De zelflerende en zelf-explorerende tools van vandaag

De mogelijkheden om machines zelf te laten leren zijn de laatste jaren erg sterk toegenomen. Zo keek de wereld geschokt toe hoe het zelflerende AlphaGo de Koreaanse wereldkampioen van het spel Go overtuigend versloeg, tien jaar eerder dan verwacht. En ook met poker heeft een computer al veel geld gewonnen van top-spelers. De makers van AlphaGo, Google's 'Deepmind', heeft ook een ander programma dat uitblinkt in zelfleren en zelf-exploreren. Dit programma slaagt er zonder enige uitleg in om klassieke Atari spellen te onderzoeken, te leren en

vervolgens tot in perfectie te spelen. Als voorbeeld kiezen we het spel 'breakout' waarvan je hieronder een screenshot ziet.



Het Deepmind-programma start het spel en heeft in het begin geen idee wat er gebeurt. Het concept van een bewegend balletje dat moet worden teruggekaatst is onbekend. Dat geldt ook voor het idee dat de roze pixels onder in beeld samen een object vormen dat te besturen is en dat de punten gescoord worden door de gekleurde blokken weg te spelen. Alles is nieuw, dus alles moet verkend en geleerd worden. Dit verkennen en leren gebeurt door het spel keer op keer te spelen.

Na tien minuten spelen krijgt het programma al enige handigheid in het besturen van de paddle. Na driehonderd spellen kan het programma net zo goed spelen als een menselijke speler dat zou kunnen. En na vijfhonderd spellen heeft het programma de optimale tactiek gevonden om het spel te winnen, door via de zijkant een opening te maken en de bal boven de hoogste rij blokken te laten stuiten. Met precies dezelfde tool kunnen ook andere spellen worden geleerd. Op Youtube vind je voorbeelden van Pacman en Super Mario. Zonder uitleg. Met één en hetzelfde tool.

Zelflerende en zelf-explorerende testtools nemen ook het denkwerk over

Stel nu dat we deze Deepmind-tool geen Atari spel zouden voorschotelen, maar een object dat we willen testen, zoals een website. Wat zou zo'n tool kunnen doen? Wat zouden we ermee kunnen bereiken?

Laten we ons nu vooral richten op wat op korte termijn toepasbaar is: invoercontroles, opmaakcontroles, foutafhandeling en hier en daar de samenhang van waarden in twee velden. Ofwel de syntactische en semantische testen.

De tool zou zonder enige voorkennis gaan klikken, gaan typen, gaan slepen enzovoort. Het zou invoervelden ontdekken en daar allerlei soorten waarden gaan invoeren. Het zou willekeurige combinaties gaan maken van handelingen en waarden en zou ervaren wat dit voor resultaat oplevert. Het zou trends en patronen ontdekken en daarmee bepalen wat standaard gedrag is. En zodra de tool weet wat de standaard is, dan weet het ook wat afwijkend gedrag is. En dan kan de tool een rapport opleveren met een lijst van deze afwijkingen. Dat is het punt waar wij als menselijke tester voorlopig nog nodig zullen zijn: het interpreteren van de afwijkingen en het onderscheid maken tussen gewenste afwijkingen en daadwerkelijke fouten. Uiteraard zal het zelflerende programma de volgende keer rekening houden met deze menselijke inbreng. Het zal de volgende testronde met een veel kortere lijst van door ons te interpreteren afwijkingen komen.

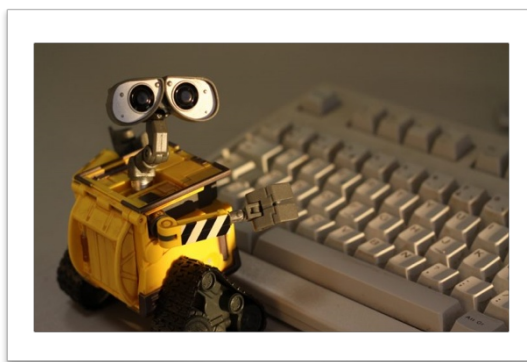
Net als bij de huidige testautomatisering zit de kracht van dit concept in het herhalen van testen. In de volgende oplevering van het testobject zal de tool herkennen wat de veranderingen zijn en maakt opnieuw geen onderscheid tussen goed en fout. De menselijke tester kan dus zowel de gewenste veranderingen als eventuele regressiemeldingen beoordelen. Trouwens, wie zegt dat dit een professionele tester moet zijn? We kunnen ook andere betrokkenen zoals de klant laten beoordelen, met steun en advies van de kunstmatige intelligentie.

Kunnen we de tool nog ergens mee helpen?

Hierboven ging het nog om de syntactische en semantische testen, maar de stap naar het automatisch testen van logische verwerkingen en procesflows is heel goed denkbaar. Zeker als we de tool een beetje op weg helpen door verschillende 'inputs' mee te geven. In principe zou de tool, na duizenden uren exploreren en leren, (bijna) alles zelf kunnen bedenken. De belangrijkste reden om inputs mee te geven is dus efficiëntie.

- We zouden testdata mee kunnen geven, zoals correcte postcodes en bestaande adressen, het formaat van telefoonnummers en bestaande gebruikersnamen en wachtwoorden.
- We kunnen requirements en ontwerpen aanleveren, het liefst in een gestructureerde vorm. De tool zal dan proberen te herkennen welke invoer- en uitvoerwaarden er in deze testbasis staan en proberen dit te valideren in het testobject.
- We zouden, na een eerste testronde, kunnen aangeven welke van de ontdekte functionaliteiten we het belangrijkste vinden, zodat de tool zich daar de volgende testronde meer op richt binnen de gegeven tijd.

Daarnaast is er nog een hulpmiddel dat wij als professionele testers graag willen toevoegen: testontwerptechnieken. Hiermee kunnen we sturen op een bepaalde dekking. Dit soort technieken schieten er nu in de praktijk vaak bij in, dus het zou veel helpen als een tool hierin kan faciliteren. Bovendien: doordat de tool in principe een oneindig aantal testen kan doen, kan AI ook zelf tot nieuwe technieken komen, misschien zelfs beter dan wij als mensen kunnen bedenken. Denk maar aan de hierboven beschreven truc die de tool zelf bedacht om het spel



Breakout het meest efficiënt te spelen. De beperking is op dit moment vooral de tijd die de tool en het testobject nodig hebben om de vele mogelijkheden uit te proberen.

Kan de tool zichzelf verder helpen?

Naast de menselijke 'inputs' is er nog een héél interessante 'input': de eerdere testen van de tool zelf. Op die manier kan de tool véél efficiënter testen tijdens een volgende testronde. Vergelijk het met het vijfhonderdste spelletje Breakout. De ervaring van de tool hoeft zich natuurlijk niet te richten op één testobject. Als we bijvoorbeeld een website hebben getest en we hebben als mens aangegeven welk soort afwijkingen we wel of niet wenselijk vinden, dan zal de tool bij de tweede website deze ervaring meenemen. Denk hierbij aan voorbeelden als gebruiksvriendelijkheid en toegankelijkheid voor bijvoorbeeld kleurenblinde gebruikers. En als we na een tijdje honderden websites hebben laten testen (of eigenlijk exploreren) en beoordelen (leren), dan ontstaan er algemene trends in verwachtingen waar de tool bij de volgende test rekening kan houden. Kennis en ervaring over honderden testobjecten, beschikbaar met één druk op de knop, is dit het einde van de professionele functioneel tester die we nodig hadden om zijn kunde en zijn jarenlange ervaring?!

Hoe haalbaar is dit concept?

Laten we weer even op adem te komen. Op dit moment zijn zelflerende en zelf-explorerende testtools nog fictief. Maar alle benodigde kennis en technieken bestaan op dit moment wel. Een belangrijke vraag is uiteraard of de investeringen in dit soort testtools opwegen tegen de resultaten. Met name het overnemen van het creatieve denkwerk van de menselijke tester klinkt nu nog heel futuristisch, maar de ontwikkelingen gaan erg snel. De eerste stap zal zijn dat de testtools een deel van het verkennende werk doen en daarmee een basis leggen waarop de menselijke tester kan voortborduren. Maar voor de langere toekomst is het niet denkbeeldig dat dit soort zelflerende en zelf-explorerende tools op eigen initiatief de kwaliteit van een informatiesysteem vaststellen op een wijze die de betrokkenen voldoende zekerheid en vertrouwen geeft om het informatiesysteem in gebruik te nemen.

Wil jij samen met ons de toekomst verkennen?

We willen graag verder met het concept, samen met collega testers die hier graag over mee willen denken. De mogelijkheden lijken eindeloos. We denken al groot en dan nog zullen we in de toekomst waarschijnlijk verbaasd worden. Maar er zijn ook genoeg obstakels om te overwinnen. Over dit kijkje in de toekomst is genoeg te bespreken! Heb je hier interesse in, laat het dan aan één van ons weten.

Verder is het leuk om te weten dat de code van het Deepmind programma om Atari spellen te spelen, publiek beschikbaar is (via GitHub). De programmeertaal is C++, dus mocht je hierin thuis zijn en je wilt iets bijdragen aan dit experiment, dan horen we graag van jou!

Het Deepmind programma dat Breakout speelt - <https://www.youtube.com/watch?v=V1eYniJ0Rnk> ←

REACTIES OUD-LEDEN OVER 20 JAAR TESTNET

Voor dit voorjaarsmagazine hebben we reacties gevraagd aan oud-leden over hun lidmaatschap van TestNet, met vragen als:

- *Wat heb je eruit kunnen halen?*
- *Wat heb je toegevoegd?*
- *Herinner je je een leuke of bijzondere ervaring over een bijeenkomst?*
- *Ben je beroemd geworden met een publicatie in een uitgave van TestNet?*



Tim Koomen • info@timkoomen.nl

'TestNet is al twintig jaar mijn belangrijkste netwerk van testers, voor mij is elke bijeenkomst weer een warm weerzien met vele collega's. En met al die gesprekjes én de vaak inspirerende presentaties en workshops ben ik vele malen geholpen in mijn werk, zowel test-inhoudelijk als in de netwerk-sfeer. Hopelijk heb ik ook af en toe anderen kunnen helpen...

Strak georganiseerde, luxe netwerk-evenementen met altijd voldoende inhoud voor zowel newbies als vakidioten, ga zo door!!!'

Felipe Ruiz Sanchis • felipe.ruiz.sanchis@nl.abnamro.com

'Bijna twintig jaar geleden was het testersvak zeer onderbelicht en ondergewaardeerd. Eerlijkheidshalve was de professionaliteit ook niet overal even hoog.

Destijds kwam een groep testers, waaronder ik, bij elkaar in Apeldoorn bij de Belastingdienst om te praten over de mogelijke vorming van een testcommunity/vereniging. Vanuit de overtuiging dat kennisdeling met vakgenoten een goede zaak is.

Ik weet nog goed dat er al snel simpelweg door handopsteken gekozen werd om een testvereniging op te richten met als eerste voorzitter Martin Pol, mede schrijver van een boek dat velen van ons als leidraad hebben gebruikt: TMap.

De Belastingdienst was onze gastheer en verzorgde ons vanuit de bedrijfskantine een simpele maar voedzame maaltijd; Nasi! Flyers werden uitgedeeld zodat je je kon inschrijven als lid. Ik gaf de mijne aan iemand anders die de flyer niet gezien had. Daardoor duurde het nog wel even voordat ik alsnog lid werd. Wat een verschil met de huidige gang van zaken: twee maal per jaar een seminar met interessante sprekers en uitstekende catering. Elke maand avondseminars met eveneens uitstekende verzorging en nog veel meer. Een vereniging met duizenden leden waar netwerking alle ruimte krijgt met een mooie site met waardevolle informatie waar je je simpel als lid kunt opgeven.

Ik heb mogen meedoen in een werkgroep voor de totstandkoming van een white paper over kwaliteitsregie. Met een groep professionele maar ook eigengereide mensen een complex stuk in elkaar componeren en aan de test-community presenteren was een hele toer. Maandelijks kwamen we bij elkaar ergens in Nederland, werd er pizza of Chinees besteld en ontstonden heftige discussies. Leden haakten af en nieuwe experts kwamen er bij. Uiteindelijk lukte het om een stuk te schrijven waar alle werkgroepleden achter stonden. Als groep komen we nog steeds één keer per jaar samen en gaan we ergens uit eten. Dat zegt iets over de positieve sfeer die er uiteindelijk in de groep ontstond.

Vele gezichten van het eerste uur kom je nog steeds tegen bij TestNet bijeenkomsten. Sommigen zijn inmiddels met pensioen. Velen zijn nieuw en dat is goed! Nieuwe mensen die in staat zijn onze niet commerciële vereniging in de lucht te houden als lid of als vrijwilliger. Een vereniging waar we met zijn allen trots op kunnen zijn. Hulde voor al die vrijwilligers die het door de jaren heen voor ons mogelijk gemaakt hebben.'

Jan Jaap Cannegieter • j.cannegieter@squerist.nl

Wat heb je uit je lidmaatschap kunnen halen?

Heel veel! Allereerst natuurlijk heel veel kennis, ideeën, visies en dergelijke. Inhoud dus en dat moet mijns inziens ook de basis zijn en blijven van TestNet. Maar ook veel gezelligheid. Ik zorg altijd dat ik redelijk vroeg op meetings ben en heb ook nog wel eens aan het eind van de meeting het licht uitgedaan. Het is ook vaak heel gezellig en een mogelijkheid bekenden weer eens te zien en nieuwe vakbroeders te leren kennen.

TestNet is ook de organisatie waar ik mijn eerste stappen op de weg van spreken op conferenties heb gezet. Eerst voorzichtig een presentatie in een klein zaaltje, later workshops met tientallen mensen, het grote podium en keynotes. Inmiddels heb ik allerlei sessies in zo'n twintig landen verdeeld over vier continenten verzorgd. En dat is allemaal begonnen bij TestNet. Vind ik presenteren op TestNet dan minder nu? Integendeel.

Wat heb je toegevoegd?

Ik hoop een boel inhoud! Op diverse gelegenheden heb ik presentaties, workshops en dergelijke verzorgd, daarmee hoop ik leden aan het denken te hebben gezet of iets te hebben geleerd. Onder het motto 'delen is het nieuwe hebben'. Overigens heb ik wel die indruk, regelmatig zijn leden teruggekomen op presentaties of workshops van mij. Vaak discussieerde ik nog met andere testaholics over iets wat iemand (waaronder ik) op het podium had geroepen. Daarnaast hoor ik vaak dat ik mensen kan enthousiasmeren, blijkbaar voeg ik dat ook toe. Maar dat gaat vanzelf, daar doe ik niets voor.

Herinner je je een leuke of bijzondere ervaring over een bijeenkomst?

Na het lezen van deze vraag moest ik meteen terugdenken aan een moment na een TestNet-meeting waarbij ik tegen een ander lid aanliep, die al jaren meeloopt. Die bleek ernstig ziek te zijn geweest maar gelukkig weer beter was geworden. Dit is geen collega of projectgenoot of zo maar gewoon ook een TestNet lid waarbij sprake is van wederzijds respect. Ik had een oprecht gevoel toen hij vertelde dat hij ziek was geweest en opluchting toen ik hoorde dat hij weer beter is. TestNet is dus meer dan inhoud, testen, netwerken, het is ook een groep mensen die menselijke relaties aangaan. Vrienden is een groot woord maar er lopen toch best een aantal mensen bij TestNet van wie ik het altijd weer leuk vind ze te zien, waarvan ik hoop dat het ze goed vergaat. Als mens, naast als tester.

Ben je beroemd geworden met een publicatie in een uitgave van TestNet?

Beroemd is een groot woord, ik ben nog niet gevraagd voor 'Sterren dansen op het ijs' of 'Wie is de mol'. En als je het beperkt tot publicaties moet ik nee zeggen. Ik weet wel dat ik gevraagd ben een presentatie of keynote te doen op andere conferenties naar aanleiding van een presentatie of workshop op TestNet. Met als mooiste voorbeeld STANZ in Nieuw-Zeeland en Australië. Iemand had me eens op TestNet iets zien presenteren, hij is toen ook nog een tijdje een collega geweest. Hij is toen via een lange weg in Nieuw-Zeeland terecht gekomen en belandde daar in de programmaraad van die conferentie. En daar heeft hij de andere overtuigd dat ik daar een keynote moest doen. Zo zijn er meer voorbeelden van conferenties waar ik terecht ben gekomen na een optreden op TestNet maar dit is toch wel de mooiste, denk ik.

Jos van Rooyen • jos.van.rooyen@identify.nl

'Ik ben eigenlijk vanaf het begin lid van de vereniging. Door regelmatig de events te bezoeken leer je veel, ontmoet mensen en bereid je netwerk enorm uit. Daarnaast is het een podium om je ideeën te presenteren en te publiceren. Beide doe ik op zeer regelmatige basis.

Verder heb ik via de werkgroepen zeker veel toegevoegd aan de vereniging. De werkgroep testregie waarbij de basis is gelegd voor regievoering. Daarvoor hebben we een Whitepaper opgeleverd, presentaties en thema avonden verzorgd.

Daarnaast natuurlijk de werkgroep HBO/ academische testopleiding. Hier zijn we druk bezig om de basis te leggen voor meer aandacht voor test/kwaliteit binnen het onderwijs. De eerste resultaten zijn er al. Ons ontwikkeld curriculum is gebruikt voor een update van de bachelor of ict/science. Daarbij heeft test een prominente plaats gekregen. Op dit moment zijn we druk bezig om een minor te ontwikkelen voor softwaretesten.

Ik herinner me de keynote die we hebben verzorgd voor de HBO testopleiding. De film die we gemaakt hadden leidde tot een spontaan applaus. Een zeer bijzondere ervaring moet ik zeggen. De reacties op ons verhaal waren super leuk!

Tony Huiskamp • t.huiskamp@home.nl

'Ik ben al een behoorlijke tijd lid van TestNet. Van een redelijk klein clubje, die zijn weg als 'vakvereniging' nog moest vinden, tot een bloeiende vereniging, met veel dynamiek. En veel meer exposure!

Continu zoekende naar de trends, wat is hot, waar is behoefte aan, hoe kunnen we de leden blijven voorzien van nieuwe/interessante informatie. Een prestatie van formaat, door het bestuur, de organisatie, maar vooral de leden.

Elk jaar waren er vele interessante evenementen, door mij gretig bezocht, leidend tot nieuwe inzichten, contacten en vervolgafspraken. Ook zag je de belangstelling en omvang elk jaar weer meer toenemen, steeds professioneler werden de presentaties en de omgeving. Mijn eigen rol was die van deelnemer aan de bijeenkomsten en themagroepen en auteur voor de nieuwsbrief.

Een van de dingen die ik me nog goed herinner, is een evenement in de jaren '90 met James Bach. Wat een drive, wat een enthousiasme, wat een nieuwe inzichten in het vak testen! Dat was voor mij een openbaring en ik ging met nog meer verdiepen in de mogelijkheden van het testvak. Ik werd me meer bewust van het belang, het bestaansrecht en de awareness van testen binnen organisaties.

Ik werk nu zo'n 23 jaar in het testvak en er blijken altijd weer nieuwe inzichten, werkwijzen, technieken en methoden.

Op naar het volgende lustrum!'

Gerard Kruijf • gerard.kruijf@centric.eu

'Ik ben twintig jaar geleden lid geworden van TestNet, echt in de beginperiode van de club. Dat was, wat testen betreft, een spannende periode; het vakgebied moest zich een positie verwerven in de IT-wereld. Testen was toen niet zo vanzelfsprekend in heel veel bedrijven. Heerlijke en stimulerende discussies tijdens de bijeenkomsten in de Eenhoorn in Amersfoort vanwaar wij altijd weer met nieuwe munitie vertrokken.

Geleidelijk aan zag je dat wij testers geaccepteerd werden en dat het IT-vakgebied en het testen daarbinnen volwassener werden.

Zo'n vereniging als TestNet kun je gebruiken om kennis te halen, maar zeker ook om die te brengen en te delen met je vakgenoten. Ik heb er altijd veel aan (gehad). Ben lid geweest van verschillende werkgroepen. Heel in het begin maakte ik deel uit van de werkgroep Testmanagement. Dat was echt pionieren, erg leuk en leerzaam. En op dit moment maak ik deel uit van de werkgroep Opleiding Testen op HBO / Academisch Niveau.

Ik heb verder gepresenteerd op evenementen, geschreven in TestNetNieuws en meegewerkt aan het boekje ter gelegenheid van het tienjarig bestaan.

De evenementen probeer ik altijd te bezoeken, erg leuk om veel bekenden uit het vak tegen te komen en onder het genot van het een en ander, de wereld in het algemeen, en die van het testen in het bijzonder, te bespreken.

Hoewel TestNet een vereniging is van leden die op persoonlijke titel deelnemen, spelen de commerciële belangen indirect ook een rol. De bedrijven manifesteren zich toch altijd, zeker richting 'eindklant'. Prachtig vond ik het altijd om te zien hoe testinhuurverantwoordelijken van bijvoorbeeld KLM, Kadaster, Rabo omzoomd werden tijdens evenementen door de vertegenwoordigers van de verschillende detacheerders.

Ook de richtingstrijd vind ik zeer de moeite waard. Vroeger met name TMap versus TestFrame, tegenwoordig Context-Driven versus de rest van de wereld. Er viel en valt altijd wat te lachen.

Ga zo door, op naar de 25!!' ←

#DUURZAAM

Door Hans Vedder • h.vedder@cimsolutions.nl  @test_duurzaam



Twintig jaar Testnet is prachtig, maar hoe zorgen we ervoor dat we over tien jaar nog bestaansrecht hebben? Is het erg om het hard bevochten testvak te laten vallen of kunnen we nog steeds ervoor zorgen dat wij als testers meerwaarde hebben in een snel veranderende wereld? Duurzaam testen blijkt hiervoor het sleutelwoord te zijn, maar hoe krijgen we dat voor elkaar?

Waar komen we vandaan?

Tussen wal en schip lag ons huidige beroep. We deden niet echt mee en serieus nadenken over testen, daar was geen tijd voor (over). Bij mij in de boekenkast lacht het eerste serieuze testboek mij nog steeds tegemoet. Nog uit het pre-TMap tijdperk, maar als je het nu doorleest, is het inhoudelijk nog steeds prima en heel bruikbaar. Het is wel dit boek dat ervoor heeft gezorgd dat er meerdere literaire werkjes volgden en zo de buitenwereld er uiteindelijk niet meer omheen kon: de kunst van het softwaretesten was beschreven. Er volgden opleidingen, er kwamen meerdere stromingen in het testvak en er ontstonden testgoeroes die testminnend Nederland meenamen naar grotere hoogten. TestNet bloeide als nooit tevoren. Eénmaal geaccepteerd als apart vak, met voor elke testsoort een apart boek en bijbehorende goeroe, ging opeens de wereld waarin de tester zich bewoog, ook veranderen. En gelukkig maar, want opnieuw konden er meer boeken worden geschreven en ideeën worden geuit over hoe het testen zich zou moeten gaan verhouden ten opzichte van deze veranderingen. Maar deze keer wel vanuit een prima zelfverdiende positie in ontwikkelland, dit in tegenstelling tot de beginjaren.



Waar staan we nu?

Testen staat vandaag de dag onder druk. We zien dat het bevechten van de tester als een apart beroep toch steeds meer beperkt wordt. Een tester moet meer kunnen dan alleen maar testen. Aan de andere kant is het kwaliteitsbewustzijn bij alle andere beroepsgroepen groter geworden de afgelopen jaren. Na twintig jaar is testautomatisering dan eindelijk in heel veel organisaties aan het lukken. Dat hebben we onder andere te danken aan het groeiende aandeel van niet-functionele testen en het groeiend aantal slimme en goedkope tools. Jammer genoeg lijkt het er wel steeds meer op dat de tester een verlengstuk wordt van de machine ... Opnieuw bedenken is met tools vaak sneller dan hergebruiken. Voldoet de oude tool niet meer, dan is er wel weer een nieuwe op de markt. Snelheid (of eerder gezegd haast), korte termijn en hoge ROI staan centraal dezer dagen.

Waar gaan we naar toe?

Dat is niet zo maar in één zin te beantwoorden. We moeten in elk geval ons vak niet als leidend gaan beschouwen en denken dat wij testers wel bepalen wat er met ons vak gaat gebeuren. Welnee, dat wordt voor ons bepaald en wij moeten er juist voor zorgen dit zo goed mogelijk in te richten. Om groot te beginnen: welke kant gaat het met onze wereld op? We zien eindelijk een kanteling in de westerse wereld, want onze welvaart brengt nu echt problemen met zich mee waar we snel last van krijgen. Veel industrieën gaan over op duurzame productie. Onze ecologische voetafdruk dwingt ons wel dat te doen. Ook in de ICT wordt steeds meer duurzaam ondernomen. Ook hierin zien we weer een volwassenheidscurve: van boeken en websites over het duurzaam gebruik van kantoor-middelen tot innovatieve platformen die stimuleren in duurzame ontwikkelingen.

Wat is duurzaam testen eigenlijk?

Allereerst: wat is duurzaam eigenlijk? Er zijn een heleboel definities van duurzaam te vinden. Om samen te vatten in wat woorden kijk ik even in het puzzelwoordenboek en vind dan wat synoniemen: *Bestendig, blijvend, consistent, degelijk, durabel, eeuwig, geschikt, bestemd om lang te bestaan, hecht, houdbaar, kostbaar, lang durend, niet telkens veranderend, niet wankelbaar, onveranderlijk, onverslijtbaar, onverzettelijk, permanent, solide en stabiel*. Dat is nogal wat. Duurzaam testen zal geschoeid zijn op bovenstaande synoniemen.



Terug naar vroeger? Welnee!

Duurzaam testen is juist een grote sprong voorwaarts. In eerste instantie lijkt het erop dat we alle snelheid die we in ons beroep hebben kunnen introduceren weer helemaal kwijt moeten, maar dat is niet het geval. Waar moeten we aan denken als we duurzamer willen testen? Denk dan eens aan het gebruik van tools. Veel tools zijn gratis en daarmee in grote hoeveelheden en snelheden binnen gehaald. Ondanks dat ze gratis zijn, vergen ze uiteindelijk wel veel onderhoud. Ook zijn ze heel kostbaar geworden als er voor elke tool maar één iemand weet hoe ze te gebruiken of te onderhouden. Daarnaast is het zonde van alle energie als deze tools worden gebruikt als wegwerp-tools en bij het minste of geringste al weer naast de vuilnisbak staan. Denk ook eens aan de afvalberg van ongebruikte scripts en bestaande testware. Daarnaast kan worden stilgestaan bij de producenten van testtools: in hoeverre geven zij aan zich druk te maken over een duurzaam voortbrengingsproces?

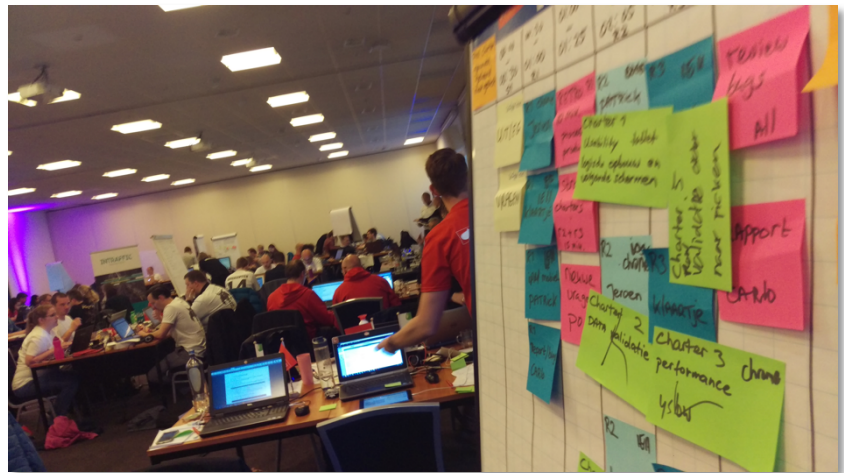
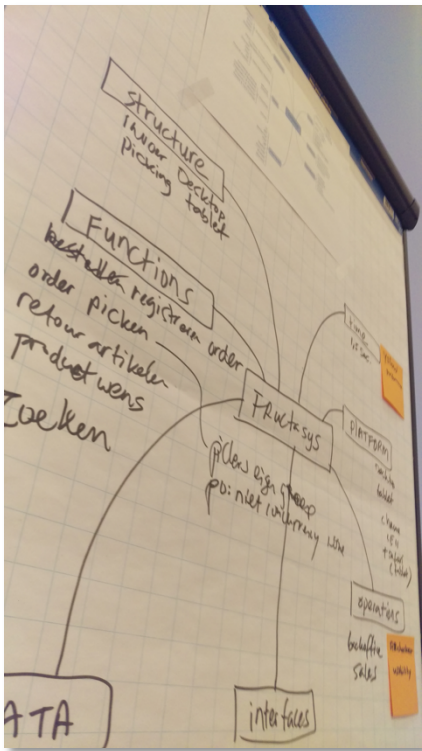
Wat ik ook veel bemerk, is dat er vandaag de dag onbepert geautomatiseerd wordt. Gewoon, omdat het kan. Maar is het ook nodig? In onze beginjaren was het een kunst om met zo weinig mogelijk testcases toch een zo veel mogelijk te kunnen testen. Nu is het de tool die onbepert kan testen en wordt er heel gemakkelijk dubbel en onnodig getest. Dat heeft ook zijn weerslag op de meerwaarde van onszelf. Laat de tester weer teruggaan naar zijn echte ambacht en ervoor zorgen dat de goede dingen worden getest. Een echte tester leeft met het te testen product mee. Een machine of tool kan dit niet en zal ervoor zorgen dat op lange termijn er een breuk ontstaat tussen juist opgeleverde producten en goed bruikbare producten.

Geen vervanging, maar juist een aanvulling!

Kijken we met een schuin oog naar het Biologisch Testen (najaarsevenement 2014), dan zien we dat 'Denk Duurzaam' één van de kernwaarden daarvan is. Verder moeten we ook denken aan het plezier en de betrokkenheid van ons als tester in de wereld van de ICT. Wat is duurzamer dan goed gemotiveerde medewerkers, die voldoende eigen bewegingsruimte hebben om hun expertise goed te kunnen uitvoeren? We moeten ervoor zorgen dat we niet tot 'alleskunnners' worden gemaakt, waarbij er roofovername op ons kunnen worden gepleegd. Het roer hoeft niet volledig om, maar zie duurzaam testen als een broodnodige aanvulling binnen het testvak, waarbij wij een heldere balans kunnen maken tussen kort gewin en langdurig rendement. Dat allemaal op het gebied van energiegebruik, testtoolgebruik, hergebruik van testware, testplezier, de afvalberg van ongebruikte



libraries en scripts, et cetera. Zoals in het begin deze eeuw het woord testfabriek nog algemeen geaccepteerd was, zal dit over tien jaar het geval niet meer zijn en zullen we allemaal spreken over scharrelsoftware, circulair testen en veganistisch scripten. Daar ligt de uitdaging voor ons de komende jaren. Succes gewenst! #duurzaam. ←



Mindmaps en Scrum boards op het NK-Softwaretesten

DE HEURISTIEK VAN TESTVERBETERING

Door Kees Blokland • kees.blokland@polteq.com



Velen kennen Test Process Improvement (TPI) van het gelijknamige maturity model met key areas en checkpoints. Aan de hand van TPI en vergelijkbare modellen is in binnen- en buitenland het testproces van legio organisaties in kaart gebracht. Als men daarbij voornamelijk 'scoort' aan de hand van de letterlijke tekst van de checkpoints gaat men volledig voorbij aan het probleem dat daardoor wezenlijke informatie gemist wordt. Lees verder en maak kennis met de heuristiek van testverbetering uit vier casussen.

Casus 1: het TPI-boek blijft ongeopend

TPI toepassen vind ik een van de coolste onderdelen van het testvak. Mijn TPI-leermeester was Martin Pol (die aan de wieg van TPI stond). Met TPI-original in de hand liepen Martin en ik drie weken rond bij twee R&D afdelingen van Huawei in Shenzhen. Een *life changing* experience en niet alleen vanwege het heerlijke Chinese eten... Onwennig zat ik met het TPI boek in de hand en een schrijfblok voor me te wachten tot het eerste interview zou starten. Iedere geïnterviewde zag vijf mensen om zich heen: twee Chinese 'TPI-experts', een vertaler, Martin en ik. Martin begon met vragen stellen en de informatie kwam los. Ik pende alles mee. Na het derde interview lag het TPI boek nog ongeopend op tafel. Huh? Mijn schrijfblok raakte echter in hoog tempo vol. We hadden na de eerste dag zelfs al een grof idee waar de problemen of verbetermogelijkheden konden zitten... Maar zonder boek?



Inmiddels weet ik hoe belangrijk het is om met open vragen gesprekken tijdens interviews op gang te krijgen. In de achtergrond spelen checkpoints van een model wel een rol, maar dan meer als 'charter' voor een interviewsessie dan als afvinklijstje. Je kunt zo al vrij snel 'beet' hebben bij het zoeken naar problemen (of positief gesteld: verbetermogelijkheden). Als dat gebeurt, dan ga je dat meteen checken (vinden anderen het ook?) of betwisten (spreken mensen het tegen als je het omgekeerde beweert?). Dit is een mooi voorbeeld van hoe vervolgvragen afhangen van de informatie die tot zo ver is verzameld: net als bij exploratory testen.

We hebben inmiddels enkele belangrijke heuristieken van testverbetering voorbij zien komen:

- *Open vragen stellen in interviews levert meer en betere informatie op;*
- *Checkpoints gebruik je niet te letterlijk, want dan beperken ze je zicht;*
- *TPI research is exploratory;*
- *De eerste contouren van verbeterpotentie ontstaan al vrij snel in het begin.*

Casus 2: TPI Next toegepast in passende context

De mate waarin een model belangrijk of noodzakelijk is, hangt af van de context. Een grote, vrij formele organisatie die TMap Next had uitgerold, wilde met TPI Next meten hoe succesvol de verschillende afdelingen waren met het implementeren van de verbeterde testaanpak. Ons team met *improvement architects* voerde een assessment uit in een aantal afdelingen. Wanneer we onzeker waren over de consequente beoordeling in de afdelingen, pakten we het TPI Next-boek erbij en dan baseerden we daar een specifieke heuristiek op, in de context van de organisatie. Bijvoorbeeld: wat zijn hier de 'relevant stakeholders'?

De heuristiek 'na drie gesprekken weet je het soms al', faalde trouwens volledig bij collega Wim ten Tusscher: 'Kees, ik dacht dat ik het op een rijtje had, maar na het interview van zonet snap ik er niets meer van'. Hij liep tegen 'de fase van de grote verwarring' aan: nieuwe informatie is compleet strijdig met het beeld dat was opgebouwd. In veel gevallen is dan sprake van een verschil van inzicht tussen mensen in de organisatie (dat is sowieso een bevin-ding). Soms verdwijnt de verwarring niet: een teken van een chaotisch werkende organisatie.

Heuristieken uit deze casus:

- De context is sturend bij het kiezen van een aanpak of model voor testverbetering;
- Te snelle conclusies leiden soms tot de fase van de grote verwarring;
- Modellen zijn nuttig bij het consistent benaderen van meerdere afdelingen;
- Modellen zijn nuttig bij het herhalen van onderzoeken naar testverbetering.

Casus 3: De context is onverbiddelijk

Als de aanpak niet past in de context, gaat het fout. Dat heb ik aan den lijve ondervonden bij het project van casus 2, bij een afdeling die al volledig Agile werkte. Modellen bieden een logica voor de volgorde waarin verbeteringen het best kunnen worden gedaan – in bijpassende context. TPI Next past goed bij organisaties die testen als aparte discipline hebben georganiseerd en langs die lijn verder willen. We hadden conform de opdracht van de organisatie bij alle afdelingen het advies op basis van TPI Next opgesteld.

Dus ook bij de Agile afdeling. Je zag aan de vraagtekens in de ogen van het publiek bij de eindpresentatie dat men in verwarring was: ze konden de adviezen niet goed plaatsen in hun context, het voelde niet goed. En ze hadden gelijk. Wij adviseerden de organisatie om de volgende keer voor Agile afdelingen TI4Agile (*) te gebruiken, dat volgens hetzelfde format is opgebouwd als TPI Next

Aandachtsgebieden		Beheerst			
1 Opdrachtgeverschap	↗	7	7	6	7
2 Mate van betrokkenheid	↗	7	7	7	7
3 Teststrategie	↗	7	7	3	5
4 Testorganisatie	↗	7	5	7	7
5 Communicatie	↗	6	7	4	6
6 Rapportage		4	7	7	
7 Testprocesbeheer	↗	7	7	6	7
8 Begroting en planning	↗	6	6	7	7
9 Metrieken	↗	2	1	2	
10 Bevindingenbeheer	↗	5	5	5	3
11 Testwarebeheer	↗	6	3	7	3
12 Toepassing van de methodiek	↗	6	7	7	
13 Testerprofessionaliteit	↘	5	4	7	4
14 Testgevalontwerp	↗	7	6	3	
15 Testhulpmiddelen	↗	7	7	5	
16 Testomgeving		6	6	5	5

Samengesteld resultaat 7 afdelingen

Omdat we de afdelingen op vergelijkbare manier beoordeelden, konden we de resultaten over elkaar heen leggen en zo de hotspots van verbetermogelijkheden zichtbaar maken. (zie figuur). De plaat laat ook het resultaat zien van een herhaling van de TPI Next-meting. Doordat we dezelfde aanpak toepasten, maakten we de groei ten opzichte van de eerdere meting op een geloofwaardige manier zichtbaar.

Key area	Forming				Norming				Performing			
1. Stakeholder commitment	1	2	3	4	1	2	3	4	1	2	3	
2. Planning & Estimation	1	2	3	4	1	2	3	4	1	2	3	4
3. People	1	2	3	4	1	2	3	4	1	2	3	
4. Interaction	1	2	3	4	1	2	3	4	1	2	3	4
5. Teamwork	1	2	3	1	2	3	4	1	2	3	4	
6. Environments	1	2	3	1	2	3	1	2	3	4		
7. Test process	1	2	3	1	2	3	4	1	2	3		
8. Test management	1	2	3	4	1	2	3	1	2	3		
9. Test profession	1	2	3	4	1	2	3	4	1	2	3	
10. Test automation	1	2	3	1	2	3	4	1	2	3		
11. Regression & E2E testing	1	2	3	4	1	2	3	1	2	3	4	
12. Defect management	1	2	3	1	2	3	4	1	2	3		

TI4Agile

en volledig past is bij de Agile context. (Ons advies werd later opgevolgd.)

Heuristieken:

- *Opnieuw: de context is leidend bij het maken van de keuze voor het model en de verdere aanpak;*
- *Modellen geven richting aan een logische volgorde voor testverbetering.*

Casus 4: Model-less

Zonder model voorafgaand aan een testverbetertraject beginnen, kan ook een goed idee zijn. Met collega Ruud Teunissen heb ik een dergelijk traject uitgevoerd. We begonnen gesprekken met een open agenda, om te begrijpen hoe diverse stakeholders de testafdeling zagen. Door hun mening te vragen, kregen we veel waardevolle informatie over wat er goed liep en wat niet. Onze brede ervaring met testen, de IT en IT-organisaties gaf ons het vertrouwen dat we in staat waren om de opdracht tot een goed einde te brengen. Aan de andere kant voelden we de onzekerheid: dekken we wel de juiste topics af, missen we niet iets belangrijks? Blijven we niet te veel aan de oppervlakte?

Key area	
1	Automation strategy
2	Automation architecture
3	Automation standards
4	Automation scripts
5	Team
6	Test environments
7	Test data
8	Tooling
9	Tool integration
10	Planning & estimation

Key areas TI4Automation

We besloten onderweg om:

- Met TI4Automation (*) de testautomatisering in kaart te brengen (we hadden ontdekt dat management veel belang hechtte aan testautomatisering);
- Met TPI Next de ketenacceptatietest te analyseren (uit respect voor de specialisten van de testafdeling: hun gedegen werkproces werd hiermee goed zichtbaar).

Een greep uit de heuristiek uit deze casus:

- *Als je niet vooraf een model kiest, blijf je flexibel om verschillende benaderingen te kiezen (dus opnieuw: de context stuurt je keuze);*
- *Het zichtbaar maken van sterke punten draagt bij aan de buy in voor een vervolg waar veranderingen van mensen worden verwacht;*
- *Research naar testverbetering gaat beter met zijn tweeën (sparren, elkaar uitdagen en aanvullen).*

(*) zie <https://improvement.polteq.com> ←

TWINTIG JAAR TESTNET: NIET ALLES IS ANDERS

Door Jan Jaap Cannegieter • j.cannegieter@squerist.nl  @jjcannegieter



Twintig jaar TestNet, dat is best een lange tijd. Ik herinner me nog goed dat ik bij de oprichtingsvergadering van TestNet in Apeldoorn was. We wilden een breed fundament leggen voor de toekomst van ons testvak. Sindsdien heb ik dit vakgebied niet meer verlaten, al heb ik in die periode ook veel andere dingen gedaan zoals requirements opstellen en managen, quality assurance, regievoering, lijnmanagement en procesverbetering. Laatst hoorde ik dat het thema van het jubileum voorjaarsevent op 15 mei 'Toekomst' is. Dan denk je: 'Ons vak heeft al dik twintig jaar toekomst. Is er dan in wezen niets veranderd?'

Testen in het verleden en heden

Toen ik laatst in een erg taaie file stond, moest ik even terugdenken aan die twintig jaar. Dan beseft je dat er toch wel heel wat veranderd is ons vak. Toen ik begon als tester werkten we nog in aparte testteams naast het ontwikkelteam, naast het ontwerpteam, naast noem-maar-op. Mijn eerste grote opdracht, de implementatie van de eerste testtool bij de Belastingdienst, was destijds een noviteit! Nu zijn er ongelooflijk veel tools die niet meer weg te denken zijn uit testtrajecten. Twintig jaar geleden was het aantal boeken over testen op één hand te tellen. Nu kun je er een enorme boekenkast mee vullen. Testconferenties waren er nog niet. Nu kun je iedere week wel ergens in de wereld naar een testconferentie. En toch, terwijl ik daar op de A28 in de file stond, besepte ik ook dat een aantal dingen in die twintig jaar níet is veranderd. Misschien zijn die dingen wel het meest interessant, omdat ze mogelijk ook iets over de toekomst zeggen. Laat ik je meenemen naar mijn huidige project.

Op dit moment ben ik in opdracht bij een non-profit organisatie van zo'n tweehonderd medewerkers. Het centrale systeem dat het primaire proces ondersteunt, is ongeveer twintig jaar oud en moet dus nodig vervangen worden. Daar is deze organisatie al ruim tweeënehalf jaar mee bezig. Dan zeg ik: 'Dit project loopt niet echt goed'. Daarom is de projectaanpak herijkt en in dat kader ben ik bezig met een quick scan op het testproces om op basis daarvan de testaanpak tot aan de implementatie te bepalen. Bij het bespreken van de eerste contouren van die testaanpak verzuchtte de CIO: 'Dus de testaanpak die we in gedachten hadden moeten we dus loslaten?' Die testaanpak bestond eruit dat de gebruikers het systeem na oplevering zouden testen. Een inmiddels bekende misvatting. Ik heb de CIO toen uitgelegd wat een product risk assessment is, wat testontwerptechnieken zijn, wat dekingsgraden en diepgang van testen betekenen. Er leefde bij hem nog een andere misvatting: dat je pas kunt testen als alle systemen in de keten zijn opgeleverd. Ik heb toen uitgelegd wat stubs en drivers zijn. Natuurlijk kwam de vraag: 'Kunnen de ontwikkelaars dan niet testen?' Toen heb ik uitgelegd dat de slager zijn eigen vlees niet mag keuren en dat je voor testen naast bepaalde kennis en kunde ook bepaalde skills nodig hebt. Eigenlijk heb ik hem uitgelegd dat testen een vak is. Een vak in continue ontwikkeling. Natuurlijk zetten we gebruikers in bij gebruikersacceptatietesten, maar functioneel testen moet je toch door professionals laten doen. Ik beseft eens te meer: bepaalde misverstanden rondom het testvak zijn nu eenmaal hardnekkig.

Heeft het testvak toekomst?

Sinds ik in het testvak ben gestapt is testen al meerdere keren dood verklaard. In mijn eerste testjaren kwam Rapid Application Development op. Daarin zou geen plaats meer zijn voor testers. Vervolgens kwamen er vierde generatietaal ontwikkeltalen. Ook die zouden testen overbodig maken. We hebben ook Model Based Ontwikkelen gehad. Alweer een einde van het testvak. Testtools zouden er ook voor zorgen dat je geen (of in ieder geval min-

der) testers nodig had. Ik zou bijna Test Driven Development vergeten, waardoor de ontwikkelaar zo goed zou gaan testen dat de tester overbodig zou worden. Ook bij de opkomst van Agile heb ik heel vaak gehoord dat er geen testers meer nodig zouden zijn. En ongeveer drie maanden geleden zat iemand met sneakers en puistjes me uit te leggen dat ik maar beter een ander vak kon leren omdat DevOps de nieuwe standaard wordt. Vaak ga ik op dit soort opmerkingen niet meer in. Ik weet inmiddels wel beter: in tegenstelling tot de vele testdoodsvoorspellingen in de voorbije twintig jaar is het testvak juist enorm ontwikkeld en gegroeid. Mijn overtuiging is dat 'testen here is to stay': er zijn namelijk dieper liggende redenen waarom testen altijd onderdeel van systeemontwikkeling zal blijven.

De belangrijkste reden dat testen nog veel toekomst heeft zit hem er naar mijn overtuiging in dat testers problemen op een andere manier benaderen dan ontwikkelaars. Ontwikkelaars hebben een sterk probleemoplossend vermogen. Testers gebruiken daarentegen juist hun kritisch denkvermogen. Alleen met de combinatie van beide benaderingen lever je een goed product op en daarom heb je professionele testers nodig. Daarnaast maken we tegenwoordig behoorlijk complexe producten. Om een beeld te krijgen van de kwaliteit van dergelijk producten moet je meer doen dan even rondklikken. Dat vereist het toepassen van testontwerptechnieken, dekkinggraden, noem maar op. Natuurlijk, tools kunnen ons werk zeker effectiever en efficiënter maken (en vaak ook leuker), maar tools zijn niet zo creatief (of zal ik zeggen: 'grillig of onvoorspelbaar') als het menselijk brein. Dus moeten we óók onze hersenen blijven gebruiken om te zorgen dat die tools de goede dingen doen. Het is dus een combinatie van kennis, kunde en skills waarmee we waarde toevoegen aan projecten. En ik verwacht dat dit zo blijft.

Dit betekent wel dat we onze basis goed op orde moeten hebben. Soms kom ik mensen tegen die zichzelf tester noemen, maar geen testontwerptechnieken kennen of beheersen. Mij maakt het overigens niet uit of je dan denkt aan de TMap, de ISTQB of een andere set testontwerp-technieken, maar ik vind toch echt dat een tester er minimaal enkele moet kennen en moet kunnen toepassen. Hetzelfde geldt voor dekkinggraden. Ook verwacht ik dat een tester een productrisicoanalyse kan uitvoeren. Je moet minimaal risk based kunnen testen, maar mooier vind ik het als je value based testen kunt toepassen. Ook verwacht ik dat je als tester het verschil kent tussen checken en testen. En natuurlijk weet je wat voor soorten tools er zijn en hoe tools jou kunnen helpen. Kortom, je moet als tester wel de basis op orde hebben. Dit, gecombineerd met de juiste skills en mindset, een portie flexibiliteit en de nodige leergierigheid geeft jou als tester bestaansrecht, nu en in de toekomst.

En dat project waar ik nu voor werk? Een collega van mij, een echte testprofessional, is daar op dit moment testontwerpen en testgevallen aan het maken op basis van de datacombinatietest en gegevenscyclustest. Met behulp van SOAP-bestanden test hij de losse systemen, waarna hij de ketentest gaat doen. Ondertussen ben ik bezig met het voorbereiden van de gebruikersacceptatietest. Dit systeem wordt met behulp van Scrum ontwikkeld. Dat heeft veel effect op testen, maar een aantal basiszaken (testontwerptechnieken, risk based testen, tools) blijven altijd relevant.

Kortom

Testen is een vak, dat heeft twintig jaar TestNet wel bewezen. En een vak kun je beter laten uitvoeren door professionals: die beschikken over de juiste skills, kennis en kunde. Mijn boodschap voor elke tester is: zorg dat je die basis van skills, kennis en kunde op orde hebt en houdt. Alleen dan kun je spreken van echt vakmanschap: gebaseerd op een verleden, heden én toekomst.

En we testen nog lang en gelukkig. ←

TESTNET NIEUWS

TestNet Nieuws is een uitgave van de Vereniging TestNet, een bloeiende vereniging met meer dan 1700 professionele testers als lid. TestNet streeft de professionalisering na van het testen van IT-producten en de vergroting van de bewustwording en het belang van testen als vak apart. TestNet stimuleert het uitwisselen en uitdragen van vakkennis, ervaring tussen vakgenoten en stimuleert onderzoek vanuit zowel wetenschappelijk als praktisch perspectief. Voor meer informatie en lidmaatschap, zie <http://www.testnet.org>.

TestNet Nieuws brengt tweemaal per jaar een Special uit met artikelen over een actueel thema uit de testwereld, gerelateerd aan het TestNet Voorjaars- en Najaarsevenement.

Daarnaast verschijnt op internet de [TestNet Nieuws Weekly](#), een blog met iedere week een artikel over testen en TestNet.

