

TESTNET NIEUWS

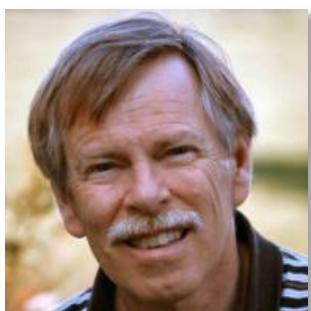
December 2012 • Jaargang 16 • Nummer 3

www.testnet.org secretaris@testnet.org



VAN DE REDACTIE

Door Hans van Loenhoud • tnn@TestNet.org [@hansvanloenhoud](https://twitter.com/hansvanloenhoud)



Creativiteit, het thema van deze TNN. Waarom denken wij testers toch altijd dat de boze buitenwereld ons niet creatief vindt? Stop daarmee! We zijn hartstikke creatief en dat kan iedereen zien die het wil zien.

Ik zeg altijd: ontwikkelaars hebben het makkelijk, die hoeven maar één goede oplossing te bedenken en daar hebben ze soms best nog moeite mee ... Wij testers, wij moeten niet alleen die ene goede oplossing bedenken, maar ook de duizend andere manieren waarop het fout kan gaan. Een rijk gevul-

de fantasie is onmisbaar als je een goede tester wilt zijn.

Okee, op zich gaan we in ons vak destructief te werk. Wij doen ons best om iets moois, een stuk software, kapot te maken. Daarmee produceren we iets anders, iets dat heel waardevol is en heel lastig te maken: vertrouwen. Ga maar eens naar de winkel en vraag of ze je een pondje vertrouwen kunnen geven. Nee toch? Vertrouwen komt te voet en gaat te paard. Vertrouwen creëren is zwaar werk, waarbij je allerlei onverwachte obstakels kunt tegenkomen. Die te omzeilen vergt inzet, doorzettingsvermogen en véél creativiteit. Zie je: zo moeilijk is het niet om uit te leggen dat testers creatief zijn. En als het kwartje dan nog niet gevallen is, laat dan deze TNN zien.

Wie dan nog twijfelt aan onze creativiteit... ←

In dit nummer

Verenigingsnieuws

Van de redactie	1
Van de voorzitter	2
Riks Column: Slimme inspecties uit Oslo	3
Interview met... Simon Gjaltema	5
Nieuws uit de werkgroepen	8
Mijn leukste of ergste testervaring	12
Boekrecensie Focus op de webbezoeker	14

Artikelen testvakgebied

Oplossingen testen is problemen zoeken	16
Een verhaal over testen	18
EuroSTAR 2012 volgens Stéphanie	22
Mappa Testi	26
Testen is kinderspel	30
Creativiteit bij testen van geluid bij telefonie	33
Testers kunnen niet innovatief zijn!	38
Evenementen en thema-avonden	43

COLOFON

Redactie

Paul Beving
Hans van Loenhoud
Kees Blokland
Johan Vink
Guido Dulos
Astrid Hodes
Rob van Steenberg
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

Opzeggen lidmaatschap: <http://www.testnet.org/algemeen/algemene-voorwaarden.html#opzeggen>

VAN DE VOORZITTER

Door Michiel Vroon • voorzitter@TestNet.org



Nu we richting de feestdagen gaan, is het de gewoonte terug te kijken naar het afgelopen jaar. Wanneer ik dat voor TestNet doe, zie ik een mooi lustrumjaar waarin we als vereniging weer het nodige hebben georganiseerd voor de leden. Zo zijn er natuurlijk als vanouds de vele thema-avonden en dit jaar hadden we het nieuwe fenomeen van de interactieve avonden. Deze opzet is zo goed bevallen dat we het continueren in het nieuwe jaar. Natuurlijk waren er dit jaar weer twee grote evenementen met daartussen de al bijna legendarische Summerschool. Het voorjaarsevenement stond dit jaar in het teken van ons lustrum en daar is dan ook ons nieuwe boek gepresenteerd. Zo'n lustrumboek begint ook echt traditie te worden binnen onze vereniging.

Het najaarsevenement met het Agile-thema kende een aantal nieuwe onderdelen die ook zeker in de toekomst terug zullen komen. Daarnaast zijn er nieuwe werkgroepen gestart zoals 'training & coaching' en 'opleiding testen'. Dit geeft aan dat we als TestNet verder willen dan alleen het organiseren van avonden en echt een platform willen zijn voor het verder ontwikkelen van ons mooie vakgebied en het actief ondersteunen van onze leden bij hun dagelijkse werkzaamheden. De website was afgelopen jaar als vanouds onze rots in de branding waar de verenigingszaken zoals aanmelding evenementen, profilering door de werkgroepen en de ledenadministratie samenkomen.

Voor het nieuwe jaar zijn de plannen weer gemaakt en deze zullen we met veel plezier aan jullie presenteren op onze ALV van dinsdag 19 maart 2013. Daar zullen we als bestuur ook de twee kandidaten voorstellen voor de vacatures die we hadden. Kijkend naar 2013 zie ik een mooi nieuw jaar met de nodige plannen en uitdagingen die het met zich meebrengt. Ik kan niet anders zeggen dan dat ik er weer zin in heb en hoop dan ook dat dat voor iedereen zo geldt.

Ik wens iedereen hele prettige dagen en een goed en voorspoedig 2013! ←

RIKS COLUMN: SLIMME INSPECTIES UIT OSLO

Door Rik Marselis • rik@marselis.eu  @rikmarselis



Eind november mocht ik, onder andere als vertegenwoordiger van TestNet, aanwezig zijn op de eerste onafhankelijke testconferentie van Noorwegen. Den Norske Dataforening (de 'Noorse NGI') heeft een testafdeling en zij organiseerden 'Testdagen ODIN 2012'. Het is me nog steeds niet duidelijk wat de Noormannen-god Odin ermee te maken had maar dat terzijde. Een van de sprekers was Tom Gilb. Dat is een Engelsman die al sinds mensenheugenis in Noorwegen woont. Velen van jullie zullen zijn naam wel kennen.

Bijvoorbeeld van het boek 'Software Inspection'. Dat publiceerde hij in 1993 samen met Dorothy Graham. Dit boek is al bijna twintig jaar hét standaardwerk op het gebied van statisch testen. Het was voor mij de eerste keer dat ik Tom in levende lijve ontmoette. Wat een gedreven man is dat!! Hij is inmiddels gepensioneerd en houdt zich nu alleen nog voor de lol met het test- en kwaliteitsvak bezig. Ik heb ruim anderhalf uur met hem gesproken. Dat wil zeggen, hij praatte en ik luisterde en stelde vragen. Hij maakte me bewust van een inzicht waarvan ik achteraf constateerde dat ik het al jaren had, maar nog nooit expliciet had gemaakt.

Tom stelt namelijk dat je niet veel inspecties hoeft te doen om het gewenste kwaliteitsniveau te bereiken. Je moet 'agile inspecties' doen. Sommige mensen uit de agile wereld maken bezwaar tegen deze term. Daarom noemt Tom het nu gewoon 'slimme inspecties'.

De gedegen en uitgebreide inspecties zoals hij in zijn boek twintig jaar geleden beschreef, hebben namelijk twee problemen. Ik vat deze even kort samen in mijn eigen woorden:

1. Imagoprobleem: inspecties kosten veel tijd en geld, 'dus dat doen we niet';
2. Effectiviteitsprobleem: inspecties halen niet alle fouten eruit.

Deze column is te kort om het imagoprobleem te gaan uitdiepen. Maar aan het eind van deze tekst zal je doorhebben dat met het uitvoeren van 'slimme inspecties' het imagoprobleem vanzelf verdwijnt.

Hoe zit het met het effectiviteitsprobleem? Want iedereen weet toch dat gestructureerde inspecties het meest effectief zijn op het gebied van vroegtijdig opsporen van fouten? Inspecties zijn inderdaad aantoonbaar meer effectief in opsporen van fouten dan alle andere vormen van statisch testen.

Maar helaas blijkt uit Toms metingen dat de beste inspecteur slechts een zesde van de major defects vindt. Uit zijn metingen blijkt ook dat het beste inspectieteam samen maar een derde van de major defects vindt. Kortom, zelfs gedurende de beste inspectie vindt men twee derde van de major defects niet. Dat maakt dat het uitvoeren van gestructureerde inspecties niet de meest effectieve aanpak voor kwaliteitsverbetering is. Je weet vooraf dat je de meerderheid van de fouten niet zal vinden.

Wat nu? Je doet een 'slimme inspectie'. Vrij vertaald is dat gewoon een steekproef. Het werkt als volgt.

Ten eerste bepaal je het gewenste kwaliteitsniveau. Bijvoorbeeld minder dan zes major defects per page. Dat is al erg ambitieus, al lijkt dat niet zo. Sla de definitie van major defects daar maar eens op na! Vervolgens inspecteer je een klein deel van je testobject. Bijvoorbeeld drie pagina's uit een ontwerpdocument van dertig pagina's. Als → je gedurende de inspectie tien major defects per pagina vindt, weet je dat gemiddeld dertig major defects per pagina aanwezig zijn. Dan is de kwaliteit dus te slecht. Je wilde immers maximaal zes major defects per pagina. In

de oude situatie zouden we de tien gevonden defects gaan oplossen. Maar daarna zouden er nog steeds twintig fouten achterblijven.

Met de nieuwe techniek gaan we de specifieke bevindingen niet oplossen. In plaats daarvan gaan we met alle betrokkenen onderzoeken wat voor soort fouten we vinden, hoe die zijn ontstaan en hoe we dat kunnen voorkomen. Kortom, we voeren een 'root cause analysis' uit en stellen daarna een 'process improvement' vast. Daarna mag de auteur het opnieuw proberen. Nadat de auteur gereed is, doen we opnieuw een slimme inspectie. Als we dan twee of minder major-defects-per-page vinden, weten we dat minder dan zes defects aanwezig zijn. Dan hebben we het gewenste kwaliteitsniveau behaald. In dit stadium kan je uiteraard ook besluiten om die één of twee gevonden majors nog wel even op te lossen.

Boeiend verhaal toch? Maar is het ook voor jou toepasbaar? Tom heeft deze slimme inspectietechniek in de vliegtuigindustrie en de defensie-industrie toegepast. Veel TestNet-leden zitten echter niet in de vliegtuigindustrie en defensie-industrie. Zij zitten in bedrijfstakken waar de 'kwaliteits-lat' minder hoog ligt. Wat heb je dan aan dit verhaal?

Ten eerste: als je een betrouwbare uitspraak over de kwaliteit van een testobject wilt doen, dan kan dat op basis van de resultaten van een steekproef. Dat is geen nieuws, maar ik begin daar toch even mee.

Ten tweede: de kern van dit verhaal is dat je niet achteraf fouten moet vaststellen, als je doel is om de kwaliteit daadwerkelijk naar het gewenste niveau te brengen. Dan blijven te veel fouten zitten. Als we constateren dat de kwaliteit onvoldoende is, moeten we direct in actie komen om het proces te verbeteren. Daar moeten we naar toe!

Misschien dat je in jouw situatie het document niet helemaal opnieuw laat maken en toch de gevonden fouten laat oplossen. Maar zorg dan wel dat je ook het proces verbetert. Op die manier krijgt het resultaat automatisch een hogere kwaliteit. Dat leidt weer tot het toenemen van het businesssucces! En zijn we gelijk van het imago-probleem van inspecties af ;-)

Op de site www.gilb.com vind je genoeg leesvoer om de hele kerstvakantie mee te vullen.

Ben je geïnteresseerd in de presentaties van de Noorse testdag (deels Noors deels Engels)? Kijk dan op <https://www.dataforeningen.no/program.242749.no.html> of tik in Google 'testdagen odin 2012' in.

Ik wens je veel inspiratie, fijne feestdagen en een succesvol 2013! ←

INTERVIEW MET... SIMON GJALTEMA

Door Rob van Steenbergem • rob@chickenwings.nl  @rvansteenbergen



Van de redactie

Met ingang van het vorige nummer van TestNetNieuws hebben we het onderdeel 'Vijf vragen aan' vervangen door 'Interview met...', zodat we echt vragen kunnen stellen die bij de persoon past. Een interview van Rob van Steenbergem met Simon Gjaltema (Simon@heesen.eu)

Hoe ben je ooit beland in de wereld van het software testen?

Eigenlijk via een grote omweg. Ik heb na mijn studie Technische Natuurkunde eerst andere boeiende werkzaamheden gedaan als algemeen coördinator van de afdeling Groningen van de Stichting Gered Gereedschap en was daarnaast voorzitter van de wijkvertegenwoordiging bij een groot renovatie-, sloop- en nieuwbouwtraject in de wijk waarin ik toen woonde. Daarna heb ik een aanvullende opleiding gevolgd op het gebied van Total Quality Management, waarbij ik als kwaliteitsmedewerker in een innovatief landmeetkundig bureau ben gaan werken. Van daaruit ben ik overgestapt naar de kwaliteitszorg in de automatisering en ben zo gaan testen. Momenteel doe ik zowel uitvoerend testwerk als testverbeteringstrajecten.

In hoeverre heb je gebruik kunnen maken van je studie (technische natuurkunde) bij het testen?

Bij natuurkunde leer je gestructureerd, logisch redeneren. Aanvullend ga je bij technische natuurkunde de theorie omzetten in een praktisch, bruikbaar resultaat. Datzelfde doe ik bij het testen; op een logische manier het te testen object benaderen en zorgen dat er een praktisch, voor de klant relevante teststrategie gevolgd wordt. Het doel is een praktisch, voor de klant bruikbaar product.

Verder heb ik bij mijn studie de programmeertalen Algol, Pascal en Fortran gebruikt. Deze kennis van programmeerstructuren bleek handig te zijn, wanneer er heel technisch opgestelde ontwerpen en delen van programma's geanalyseerd moesten worden.

Waar ben je nu mee bezig ?

Op dit moment ben ik druk bezig om een nieuwe opdracht te zoeken, bij voorkeur in het noorden van het land (dus zoek je nog een leuke collega ...). Ik woon in Groningen en helaas zit het meeste werk momenteel in het midden en het westen van het land. Qua reisafstand is dat wel ver, omdat ik het ook belangrijk vind om voldoende tijd aan mijn gezin en mijn nevenwerkzaamheden te kunnen besteden.

Ik ben momenteel secretaris van een Scouting groep. Het is heerlijk om met deze enthousiaste, actieve jongeren samen te werken! Ook zit ik in een programmagroep voor 2014 van het 4Mei-Projekt in Groningen. Dat project(koor) brengt ieder jaar weer een ander programma op basis van een thema dat aan 4 mei is gerelateerd. Dat is veelomvattend: uit te voeren werken zoeken of een compositieopdracht verstrekken, locaties zoeken, musici/solisten inhuren, subsidies aanvragen, PR verzorgen enzovoort. →



Maar even terug naar het testen: ik vind het leuk om actief de vakinhoud bij te houden, bijvoorbeeld met een TestNet Summerschool over de Best Practices van Agile werken en de (noordelijke) TestNet bijeenkomsten, het bezoeken van Microsoft voor even snuffelen aan hun nieuwe release van TestManager (dit product lijkt erg interessant voor het vastleggen van handmatig uitgevoerde testgevallen, bijvoorbeeld bij exploratory testen), het lezen van artikelen en soms wat vakliteratuur.

Wat voor vakliteratuur lees je op dit moment of heb je onlangs gelezen?

Naast natuurlijk ons eigen TestNet Nieuws lees ik meestal het Engelstalige 'Professional Tester' en de artikelen die als E-book door de EuroSTAR Software Testing Community worden uitgegeven. Dat laatste vind ik wel de meest interessante bron.

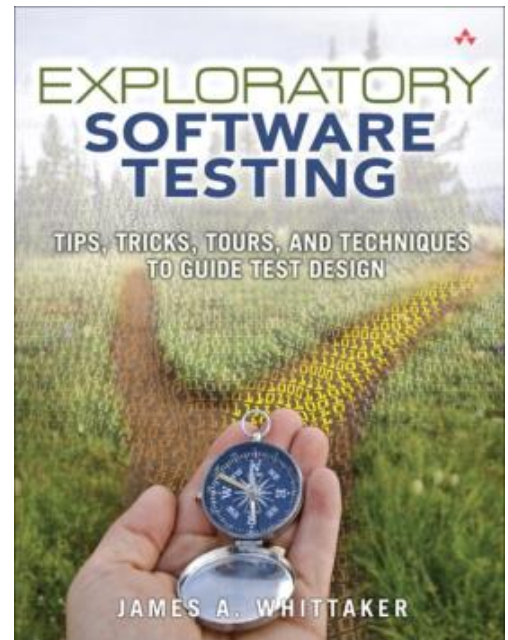
Heb je nog een tip voor testers om te lezen (artikelen of boek, anders)

Ik heb zelf net Exploratory Software Testing van James A. Whittaker weer eens ter hand genomen omdat ik daar in de praktijk nog niet zelf mee gewerkt heb en de aanpak juist binnen Agile projecten erg praktisch lijkt.

Met erg veel plezier heb ik indertijd ook 'De kracht van Scrum' van Bartosz gelezen, maar eigenlijk verwacht ik dat iedereen dit boek al lang kent.

Je weet veel van testverbeteringstrajecten. Wat is het eerste wat je doet als men je vraagt voor dit soort rollen of onderzoek?

Het belangrijkste is eerst te weten komen waarom men een testverbeteringstraject wil inzetten. Dus het achterhalen van de eigenlijke problemen. Soms blijken die niet echt bij het testen te liggen, maar bijvoorbeeld bij het gehele voortbrengingsproces. Het is dan goed om vooraf al geverifieerd te hebben dat de opdrachtgever de verbeteringsslag ook naar dat niveau wil doortrekken.



Ook moet de wil aanwezig zijn om capaciteit vrij te maken voor het verbeteren, Wanneer mensen niet vrijgemaakt kunnen worden om naast het oplossen van productieproblemen ook met verbeteringen bezig te zijn, blijkt in de praktijk dat het verbeteringstraject een stille dood zal sterven. Het belangrijkste is dat je met het bedrijf of project samen het traject trekt. Juist de interactie met de echte uitvoerenden is leuk. Ik houd van een beetje pragmatisme: 'welke informatie wil de gebruiker nu eigenlijk echt krijgen door het testplan' in plaats van het vullen van een sjabloon, even bewust kiezen voor een bepaalde testtechniek in plaats van doen zoals je het steeds gedaan hebt. Misschien is dat wel een eigenschap van een nuchtere noorderling: gewoon met twee benen op de grond blijven staan en weten waarvoor je het doet. →

Je bent al lang bezig in het testvakgebied. Zie je vanuit jouw perspectief nieuwe ontwikkelingen op testgebied de afgelopen jaren? Of is het oude wijn in nieuwe zakken?

Ik moet zeggen dat ik heel gecharmeerd ben van het Agile werken. Dat maakt weer het heel praktische in mij los: direct contact met de eindklant en meteen resultaat zien. Al betekent dat wel, dat in een groot project met veel Agile deelprojecten er toch nog wel wat afgestemd moet worden (Product Risico Analyse, interfaces tussen deelsystemen, end-to-end test). Ik heb nog niet meegemaakt dat dit op een totale Agile manier plaatsvond, maar moet ik dat op die schaal ook wel willen?

Persoonlijk beschouwde ik op het gebied van testprocesverbetering in eerste instantie TPI Next geheel als oude wijn in nieuwe zakken; Er leek alleen een klein beetje risicosturing aan toegevoegd te zijn, maar daar hield je in het algemeen toch al rekening mee. In de praktijk blijkt vooral de nieuwe wijze van presenteren (ook kunnen scoren op een hoger niveau als een lager niveau nog niet voor 100% voldaan is) erg positief ontvangen te worden door de klanten. Het was met TPI wel eens erg vervelend om na een verbeteringsslag te moeten melden: jullie zijn flink vooruitgegaan, maar op de TPI scorekaart staan jullie nog steeds op niveau 0. Omdat managers cijfertjes meestal het belangrijkste vinden, waren ze dan 'not amused'. Eigenlijk hadden ze in dat geval gewoon gelijk; verbeteren van mensen bereik je alleen met veel positieve terugkoppeling. Iedereen wil graag toch een goed resultaat kunnen zien? ←

NIEUWS UIT DE WERKGROEPEN

Door Huib Schoots • huib.schoots@gmail.com en Ray Oei • rayyho@gmail.com



Werkgroep training & coaching

Woensdag 14 november was de werkgroep training en coaching vertegenwoordigd op de werkgroepenavond. Helaas was de avond in zijn totaliteit slecht bezocht en dat is erg jammer omdat de vier werkgroepen een veelzijdige en interessant programma te bieden hadden. De werkgroep agile had twee zeer interessante verhalen en de werkgroep HBO/academische opleiding had een levendige en inhoudelijk goede discussie. Helaas hebben we het programma van de werkgroep usability moeten annuleren vanwege een gebrek aan belangstelling.



Presentatie training & coaching

Bij de presentatie van de werkgroep training en coaching waren zo'n vijftien mensen aanwezig. Na een korte uitleg over de aanleiding en de inhoud van de werkgroep werd er voor de pauze gediscussieerd over wat de aanwezigen graag zouden willen zien van de werkgroep. Na de pauze was er ruimte voor twee korte demo's van een Miagi-Do challenge en een testing dojo. Wat houden de voorgestelde werkvormen in en wat is er uit de discussie gekomen?

Wat wil de werkgroep?

De werkgroep zal zich gaan richten op, de naam zegt het eigenlijk al, trainen en coaching van testers. We willen de leden van TestNet helpen met het leren van nieuwe vaardigheden en nog beter worden in hun vak. Nadrukkelijk niet door het geven van opleidingen die al breed beschikbaar zijn (denk aan: TMap, ISTQB, agile testen, test management, enzovoort). Maar we denken aan alle andere mogelijke vormen die TestNet naast de gangbare trainingen nog kan doen om testers op te leiden en te coachen. Het doel is wat ons betreft om testers de mogelijkheid te geven meer te oefenen in een veilige omgeving. Onze boodschap aan de test community is: train, lees, daag jezelf uit en oefen!

De werkvormen

Na de eerste bijeenkomsten van de werkgroep denken we aan verschillende werkvormen.

WeekendTesters

Op een zaterdag of zondag gezamenlijk testen via de tekst-chat van Skype. De participanten oefenen met aspecten uit hun vak: afwisselend in de breedte en in de diepte. De gezamenlijke testsessie duurt maximaal één uur, gevolgd door een debriefing van nogmaals maximaal één uur. Hierin delen de deelnemers hun ervaringen, leermomenten en handigheden onderling. Mogelijk willen we dit ook door de week in de avonduren gaan aanbieden. Zie ook de website <http://weekendtesting.com/> →

TestChat

Via Twitter kunnen zowel prikkelende als inhoudelijke onderwerpen worden ingebracht om over te discussiëren. Dit biedt de mogelijkheid om met elkaar zowel kennis als ervaring te delen. In een chatsessie van ongeveer een uur worden vijf centrale vragen behandeld. Een facilitator houdt de discussie in dit uur beperkt tot de vragen die vooraf gesteld zijn. Na het uur kan de chat nog doorgaan en kan een onderwerp in detail besproken worden.



Online (Skype) Coaching

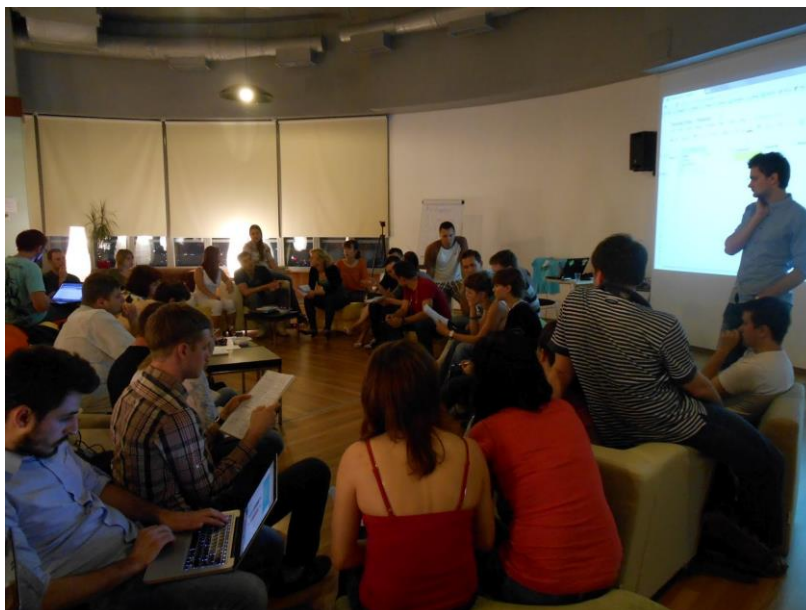
Een coachingsessie is een één-op-één interactie via de tekst-chat van Skype. Een coachingsessie is gericht op het verbeteren van bepaalde skills. Het is de tester zelf die het probleem aandraagt en tijdens de sessie ook zelf de oplossing bedenkt, uiteraard met hulp van de coach. De 'student' wordt geholpen met een socratische methode (zie: http://nl.wikipedia.org/wiki/Socratische_methode). Vaak krijgt de tester gedurende de sessie een taak of oefening. Na afloop van de oefening wordt ruim de tijd genomen om te debriefen of te evalueren. Dit naar het voorbeeld van de AST.

Zie: <http://www.associationforsoftwaretesting.org/about/membership/skype-coaching/>

Testing Dojo

Een Testing Dojo is een fysieke bijeenkomst waar de testers bij elkaar komen om met elkaar een product te testen. Voorbeelden van mogelijke opdrachten: het testen van een website of andere software, het genereren van testideeën voor bepaalde software of het rapporteren van bugs. Vaak zullen de opdrachten uitgevoerd worden op een exploratieve manier, gebruikmakend van exploratory testing. Testing dojo's bieden een veilige omgeving om zonder druk van deadlines te oefenen en van elkaar te leren. Testen dojo's zijn een manier om groepsgewijs te leren maar ook om nieuwe testers op te leiden.

Zie ook: <http://www.testingdojo.org/> →



Miagi-Do School of software testing

De Miagi-Do school of software testing is een organisatie opgericht door Matt Heusser in de VS. Het heeft als doel om testers uit te dagen te leren door challenges te doen. De school biedt testers de kans te laten zien dat ze bepaalde skills hebben en deze te verbeteren. Het principe is eenvoudig: de school bestaat uit een aantal instructeurs, die opdrachten aan de studenten geven. Door een opdracht goed uit te voeren kun je 'brown-belt' of zelfs 'black-belt' worden. De opdrachten worden gedaan via mail of social media (denk aan Skype text chat) en worden altijd achteraf uitgebreid geëvalueerd met de student. Hier vind je een blogpost van een student aan de internationale Miagi-Do School: <http://martialtester.wordpress.com/2012/02/12/miagi-do-test-challenge/>

BBST (in het Nederlands) in samenwerking met AST

BBST (black box software testing) is een vier weken durende online training van de AST (Association for Software Testing). Het materiaal bestaat uit video's ingesproken door Cem Kaner, slides en artikelen. De deelnemers moeten iedere woensdag en zaterdag iets opleveren: een proefexamen, een groepsoefening of een individuele opdracht. Alle opdrachten worden gemaakt via een online forum en een wiki. De cursusleiders zien erop toe dat de cursisten hun opdrachten op tijd (stipt om middernacht op de gegeven datum) inleveren. Het is een intensieve training op academisch niveau. Al het materiaal is gratis beschikbaar via <http://www.testingeducation.org/BBST/foundations/> en is in het Engels. De reden dat de werkgroep aan deze vorm denkt, is om het mogelijk te maken voor TestNet-leden de interactie met medestudenten en de instructeurs in het Nederlands te doen.



Discussie werkgroepenavond

De voorgestelde werkvormen zijn goed ontvangen en de deelnemers gaven aan dat ze met één of meer werkvormen aan de slag willen. Leuk!

Daarnaast werden de volgende vragen gesteld of opmerkingen gemaakt:

- Ik mis een forum om vragen te kunnen stellen en ervaringen uit te wisselen.
- Wat moet ik leren?
- Wat maakt een goede tester? Hoe ziet een testers DNA eruit?
- Ik werk in een niche, hoe vind ik mensen die hetzelfde willen leren als ik?
- Hoe weet ik dat ik 'goed' test?
- Hoe voorkom ik rituelen in mijn werk?
- Wat is het rendement van een tester? Hoe meet ik dat? En hoe verbeter ik dat?
- Kan TestNet een lijst samenstellen met interessante links, goede boeken (liefst met boekbesprekingen en/of samenvattingen) en andere zaken die testers kunnen bestuderen?
- Kan TestNet een bibliotheek maken en boeken uitlenen aan haar leden?
- Hoe selecteer je een goede tester in een intake- of sollicitatiegesprek?
- Is het mogelijk om een online Open Space bord te maken waar testers met elkaar kunnen afspreken om op vaste tijden (bijvoorbeeld bij thema-avonden in de extra ruimte) over specifieke onderwerpen te praten?

Allemaal geweldige vragen en ideeën die we als werkgroep zullen bespreken. We gaan kijken of het mogelijk is ze op te pakken. Mocht je één van de genoemde initiatieven willen helpen oppakken, stuur dan een e-mail aan werkgroepen@testnet.org →

Korte termijn

Op korte termijn zullen we starten met Test Chat, weekend testers, Miagi-Do en online coaching. Volgend jaar is er bij het NBC op iedere thema-avond een extra ruimte geboekt voor interactieve activiteiten. De werkgroep training & coaching zal, bij voldoende belangstelling, deze ruimte gaan gebruiken voor testing dojo's en andere activiteiten. Ray Oei, een instructeur van de BBST cursussen, gaf aan dat de BBST cursus in het Nederlands mogelijk is en dat er op dit moment met de AST wordt overlegd hoe we dit mogelijk kunnen maken. Meer nieuws verwachten we begin volgend jaar. Voor specifieke vragen kun je Ray e-mailen: ryho-bbst@xs4all.nl

Doe je mee?

Zin om mee te doen? Als deelnemer of misschien zelfs als organisator of instructeur van een van de activiteiten van de werkgroep? Hou de TestNet Nieuwsflits goed in de gaten! De komende maanden zullen onze activiteiten daarin aangekondigd worden. Heb je aanvullende ideeën of opmerkingen naar aanleiding van dit artikel of de discussie tijdens de werkgroepen avond? Mail de werkgroep op: werkgroepen@testnet.org. Vermeld er duidelijk bij dat het om deze werkgroep gaat. ←

MIJN LEUKSTE OF ERGSTE TESTERVARING

Door Reinier van de Pas • Reinier.van.de.pas@bartosz.nl



We hebben deze keer Reinier van de Pas gevraagd naar zijn leukste of ergste testervaring.

Wat is jouw leukste of ergste testervaring?

Mijn leukste testervaring is eigenlijk niet hetzelfde als mijn ergste testervaring. Wat ik in de afgelopen twaalf jaar in het testvak als leukste heb ervaren, is dat ik mocht meewerken aan een geheim project, waarbij een grote Nederlandse bank serviceprovider voor mobiel werd. De uiteindelijke lancering was een succesvolle verrassing voor de Nederlandse telecommarkt en de concurrenten.

Mijn ergste testervaring, waarover ik het hierna verder zal hebben, is eigenlijk geen testervaring, maar een productie-ervaring. We hadden als programma net een grootschalige conversie uitgevoerd in productie. Helaas bleek dat een voorname set van gegevens niet volledig juist was aangeleverd door bronsystemen en/of was verwerkt in ons nieuwe systeem. Gelukkig voorzag ons systeem in opschoningsfunctionaliteit, waarvan we dankbaar grootschalig gebruikmaakten. Helaas bleek toen de opschoningsfunctionaliteit op haar beurt evenmin foutloos te werken. De problemen werden daardoor eerder groter dan kleiner. Dit was één van de argumenten om een compleet nieuwe conversie uit te gaan voeren ...

Wanneer speelde dit zich af?

Deze problematiek speelde zich een aantal jaren geleden af.

Kan je ons wat meer vertellen over deze ervaring?

Gelukkig waren wij niet verantwoordelijk voor het ontwikkelen of testen van die opschoningsfunctionaliteit. Ons team maakte slechts gebruik van opgeleverde en aanwezige beheerfunctionaliteit in productie. Voor mij was het wel een vreemde gewaarwording dat er functionaliteit in productie kan staan, waarvan de correcte werking niet volledig is aangetoond en niet duidelijk is of het nu wel of niet is geaccepteerd. Gelukkig verdwenen de kinderziektes gaandeweg uit het systeem en werd de tweede conversie een groot succes, mede dankzij het leren van ervaringen uit het verleden.

Waarom is dit jouw ergste testervaring?

Vanwege de impact; het is toch wel erg om voor bijna 500.000 klanten fouten in de conversie te constateren en vervolgens in het herstel daarvan geconfronteerd te worden met nieuwe problemen. Het heeft daarnaast behoorlijk wat tijd en energie gekost om alles weer te herstellen. Hoewel het zich allemaal in de productieomgeving afspeelde, draaide het nieuwe systeem in feite alleen nog in schaduw. Hierdoor hebben de klanten zelf gelukkig niets gemerkt van de problemen. →

Wat heb je er van geleerd en wat zou je nu anders doen?

Ik zal er minder snel vanuit gaan dat in productie aanwezige functionaliteit foutloos werkt en aanraden om daar in de acceptatie nog scherper op te zijn. Bij die acceptatie hoort ook het beoordelen van testgevallen en testresultaten van eerdere testfasen en analyse van openstaande bevindingen. Daarmee moet je een goed beeld kunnen krijgen van de openstaande risico's voor productie. Verder doe je er verstandig aan, als je tóch iets wilt 'testen' in productie, om eerst een kleine populatie te nemen. Pas als dat goed gaat, kun je het opschalen naar steeds grotere populaties. Dit klinkt als een open deur of gewoon boerenverstand, maar is daarom niet minder noemenswaardig.

Aan wie geef je deze rubriek door?

Ik geef de pen graag door aan mijn oude bekende Dick Groen. ←

BOEKRECENSIE FOCUS OP DE WEBBEZOEKER

Door Johan Vink • vink.jd@gmail.com



Over Ben Vroom

Ben Vroom is eigenaar van Ben Vroom Usability. Hij test websites bij gebruikers, beoordeelt websites en geeft advies over de inrichting van websites. Hij publiceert regelmatig over usability, het testen van websites en aanverwante onderwerpen. In 2010 heeft Ben Vroom voor Testnet een lezing over usability testing gegeven. Daardoor werd onder de TestNet-leden veel belangstelling gewekt voor dit onderwerp.

Focus op de webbezoeker

Eerder schreef Ben Vroom het boek 'Websites testen bij gebruikers'. In zijn nieuwe boek is het onderwerp breder: wat kan je allemaal doen om een website zo goed mogelijk af te stemmen op de behoeften en het zoek- en leesgedrag van de beoogde bezoekers? Volgens Ben raakt dit de kern van het onderwerp waar veel leden van TestNet zich mee bezighouden: hoe voorkom je dat je te veel vanuit eigen of organisatorisch perspectief iets ontwikkelt dat vervolgens in de praktijk niet optimaal functioneert? Hoe zorg je vanaf het begin van de ontwikkeling van een website dat die vanuit het perspectief van de behoeften en het gedrag van de doelgroep wordt ingericht? Deze vraag speelt zowel bij websites als bij applicaties waarbij TestNet-leden betrokken zijn. Daarover gaat het boek 'Focus op de webbezoeker'.

Invalshoek

De invalshoek die Ben bij het schrijven van dit boek heeft gekozen is de volgende: hoe kan je met eenvoudige, relatief goedkoop in te zetten technieken de focus naar de beoogde gebruikers verleggen en zorgen dat de website op hun behoeften en gedrag is afgestemd? Daarnaast besteedt Ben veel aandacht aan de manieren waarop je de inhoud van een website goed vindbaar kan maken. Vaak wordt bij dit soort technieken aan professionele en relatief kostbare benaderingswijzen gedacht. Daardoor zijn de mogelijkheden uiteindelijk beperkt. Met eenvoudiger technieken kan je vaak meer bereiken. Bijvoorbeeld doordat je met hetzelfde budget gedurende de hele ontwikkelcyclus een combinatie van technieken kunt inzetten. Starten met een doelgroeponderzoek aangevuld met persona's. Vervolgens usability testing in de volgende stadia van de ontwikkeling van de site én bestudering van bezoekersstatistieken na de lancering.

Het boek is over het algemeen prettig leesbaar geschreven. Het is duidelijk dat de auteur behoorlijk is geïnspireerd door Steve Krug. Evenals Steve Krug is Ben duidelijk een voorstander van een eenvoudige en pragmatische benadering van Usability. →



In het eerste hoofdstuk beschrijft de auteur de manier waarop je de website toespitst op de behoeften van beoogde doelgroep. Daarbij gaat hij in op manieren waarop je de bezoekersbehoeften kan inventariseren. Daarnaast beschrijft hij de technieken om tot persona's te komen. Het is daarbij prettig dat hij de tekst verduidelijkt met duidelijke en aansprekende voorbeelden. Bruikbaar als je als tester vanaf de start bij het ontwikkelen van een website betrokken bent.

Aandachtspunten en tips

In hoofdstuk twee en drie behandelt de auteur de inrichting van de website en de invulling van afzonderlijke webpagina's. In hoofdstuk twee gaat hij kort in op card sorting als techniek om tot een navigatiestructuur te komen die de doelgroep begrijpt. Maar deze hoofdstukken laten zich voor de rest het beste samenvatten als een lijst met aandachtspunten voor de structuur en de invulling van webpagina's. Dat vind ik persoonlijk niet lekker lezen. Wel nuttige informatie die ik als tester zou verwerken in een checklist.

In hoofdstuk vier behandelt Ben het testen van de bruikbaarheid en de effectiviteit van websites. Daarbij gaat hij uitgebreid in op het belang van het zo snel en zo vaak mogelijk testen van websites door de eindgebruikers. Hij betoogt dat het testen door gebruikers nadat de website door het ontwikkelteam is opgeleverd niet efficiënt is. Daarnaast geeft Ben tips die je kunt gebruiken in verschillende aanpakken variërend van het testen door de ontwikkelaar tot het testen van de complete website door gebruikers. Het hoofdstuk gaat naar mijn mening niet erg diep in op het testen van websites. Over dit onderwerp is veel meer te schrijven. Mogelijk is dit een bewuste keuze geweest van Ben. Het is immers geen boek dat alleen over het testen van websites gaat.

Het laatste hoofdstuk gaat in op het verzamelen van feedback. In dit hoofdstuk gaat Ben in op het gebruik van feedbackformulieren, maar vooral op de mogelijkheid om Google Analytics in te zetten voor het analyseren van het succes van de website. Bijvoorbeeld ter ondersteuning van A/B testen.

Ben heeft in de opzet van het boek 'Focus op de webbezoeker' ervoor gekozen om het onderwerp breed en laagdrempelig te behandelen. Dat heeft als voordeel dat je als lezer tips krijgt die je tijdens het hele ontwikkeltraject kunt gebruiken bij het ontwikkelen van een succesvolle websites. Het heeft als nadeel dat de meeste onderwerpen niet diepgaand zijn uitgewerkt. Als je de ambitie hebt om je als tester of ontwikkelaar echt te verdiepen in het ontwikkelen van efficiënte en bruikbare websites, dan raad ik je de boeken van Steve Krug en Niels Jacobsen aan.



OPLOSSINGEN TESTEN IS PROBLEMEN ZOEKEN

Door Bert Wijgers • B.Wijgers@vgz.nl



Testen is in essentie niet creatief, maar juist destructief. Het is daarbij wel goed om al je intellectuele vermogens aan te spreken en niet alleen de gebaande wegen te bewandelen, maar het doel is nooit om iets nieuws te maken. Onze kracht ligt in het vinden van problemen, niet in het oplossen ervan. Onze primaire taak is om informatie te verschaffen en het enige dat we daarmee creëren is duidelijkheid.

De vraag of een plan goed is uitgevoerd bij softwareontwikkeling is vaak niet met alleen 'ja' of 'nee' te beantwoorden. In alle situaties waar dit wel zo is, kunnen we stellen dat testen zich beperkt tot het controleren of de software zich conform de specificaties gedraagt. Hier komt weinig intellectuele uitdaging aan te pas; deze controles kunnen we dan ook automatiseren. Het uitvoeren van regressietestgeval- len, handmatig of geautomatiseerd, is zonder twijfel de minst creatieve bezigheid in ons vak.

Proza over poëzie

Het verzinnen van deze testgevallen is ook niet heel erg creatief. Het toepassen van testtechnieken op specificaties is wel iets dat je moet leren, maar iedereen die dit onder de knie heeft, komt tot dezelfde logische testgevallen. De creativiteit in het testen moeten we zoeken in activiteiten die te maken met onvolledige of onduidelijke specificaties. Het is zelfs denkbaar dat testgevallen zelf de specificaties zijn. Dit gebeurt nog niet veel, maar het integreren van testactiviteiten in de ontwikkelcyclus is de enige manier om invloed te hebben op het moment dat software wordt gecreëerd, namelijk het moment dat de code wordt geschreven.

Als we programmeren vergelijken met poëzie, dan kun je stellen dat ontwerpen en testen beide proza zijn. Dit proza heeft direct betrekking op de poëzie; het kan niet los daarvan bestaan. Het ontwerpen van software is het opstellen van rijmschema's en het afbakenen van het onderwerp. Het is meestal onze taak om achteraf een verhaal te maken over het gedicht. In het beste geval ontwikkelen de code en de tests zich tegelijk. Het resultaat is dan een programma met één beschrijving die tegelijkertijd het ontwerp en de test is.

Kunst van de kritiek

Testactiviteiten die het ontwikkelteam helpen om werkende software op te leveren zijn positief; ze dienen om aan te tonen dat de software werkt volgens verwachting. Op een zeker moment bereikt de software een mate van volwassenheid waarin een andere vraag belangrijk wordt: 'Zijn er situaties waarin de software niet volgens verwachting werkt?' Dit leidt tot negatieve testactiviteiten die erop gericht zijn om problemen aan het licht te brengen. Negatief testen vergt een totaal ander perspectief dan positief testen, mede omdat het niet uitsluitend gebaseerd is op specificaties. Is het daarmee creatief? Integendeel, negatief testen is gericht op het laten falen van de software; het is destructief!

Door het blootleggen van zwakke punten stellen we anderen in staat om de software te verbeteren. Door het vertellen van een informatief verhaal over de software leveren wij indirect een bijdrage aan het scheppende moment. Dit verhaal moet zo eerlijk mogelijk zijn, dus we hoeven niets te verzinnen. Wel moeten we ons continu afvragen hoe betrouwbaar onze informatie is, anders bestaat de kans dat we sprookjes vertellen; heel creatief, maar ook contraproductief. →

Ons vak is niet creatief; wij ondersteunen degenen die het scheppende werk doen. Softwaretesten is wel een kunst, iets waar je goed in kunt worden, maar het is de kunst van de kritiek. Als we niet eerlijk tegen onszelf zijn, hoe kunnen we dan eerlijk zijn tegen anderen? ←

EEN VERHAAL OVER TESTEN

Door Stefan Croes • s.croes@furore.com



Stefan: 'Waar ben je naar op zoek? Wat heb je nodig?'

Maarten: 'Een verhaal. Een verhaal over hoe wij testen.'

Ik zit met mijn manager op een terras. We hebben net een afdelingsoverleg gehad en de borrel bevindt zich in de fase waarin je nog nét over werk mag praten. Het eerste biertje, de tweede is net besteld.

Het is een begrijpelijke vraag. Wij zijn een commercieel bedrijf en hij is op zoek naar een helder verhaal waaruit blijkt hoe je je als tester in de markt kan onderscheiden.

Als hij me de vraag een paar jaar geleden had gesteld, had ik hem een standaard antwoord kunnen geven. Maar hoe meer ik over testen, software, tooling, klanten en organisaties heb geleerd, hoe minder zeker ik ben dat er een eenduidig antwoord bestaat op zijn vraag. *'The more I know, the more I know I don't know'*.



TMap als antwoord?

Het makkelijkste antwoord dat je als bedrijf zou kunnen geven is dat je werkt volgens de TMap methode. Het is de geldende standaard in Nederland en het is een verhaal dat veel klanten verwachten te horen. Zoals vrijwel elke tester heb ik een TMap-certificaat en heb ik me een tijdlang vol overgave volgens deze methode op testen gestort. Na een tijd had ik het idee dat ik de methode goed onder de knie had. Uitvoeren van een risicoanalyse, bepalen van testtechnieken, opstellen testspecificaties, testuitvoer, rapportage. Als iets niet volgens het ontwerp was gebouwd, dan rapporteerde ik het. Keurig volgens het boekje. Maar... niet bij elke testopdracht die ik deed had ik tijd voor een uitgebreide testvoorbereiding. Ook beschikte lang niet elk project waaraan ik meewerkte over ontwerpdocumentatie. Hoe kan je dan testgevallen opstellen? En als de specificaties tekortschieten, hoe weet je dan of iets wel of niet werkt? Is een functioneel ontwerp eigenlijk wel de enige, of de belangrijkste bron van informatie voor testen? Het zijn vragen die ik mij al stel vanaf mijn eerste testopdracht.



De testspecificaties had ik in een paar dagen opgesteld op basis van een nogal summier ontwerp en na de oplevering was ik in één dag klaar met het uitvoeren van de tests. Resultaat: twee bugs. Maar wat te doen met de overige vier dagen in die week? Ik ben gaan onderzoeken of ik fouten kon vinden als ik dingen deed die niet in het ontwerp stonden beschreven.

Hé, dat was leuk om te doen! En opeens liep het aantal gevonden bugs snel op. Een bemoedigend resultaat voor een beginnende tester en ook een resultaat dat me aan het denken zette. Hoe kan ik er nou voor zorgen dat ik zulke fouten per definitie vind? Er zijn geen grenzen aan mijn fantasie en creativiteit om testscenario's te bedenken, maar er zijn wel grenzen aan de tijd die ik aan het uitschrijven van testgevallen wil besteden.

Om over de grenzen van het budget van de opdrachtgever nog maar te zwijgen... →

Een tweede leermoment kwam nadat ik al ruim een jaar tester was. Ik was inmiddels gecertificeerd en voelde me vrij zeker dat ik alle zaken die niet conform het ontwerp gebouwd waren kon vinden.

Na een voor mijn gevoel succesvolle testperiode werd de applicatie vrijgegeven voor een acceptatietest door de klant. Voor een zelfkritische tester een belangrijk moment; de fouten die een klant vindt zijn immers zaken die je als tester, om wat voor reden dan ook, over het hoofd hebt gezien.

Het werd een pijnlijke ervaring. De meldingen uit de acceptatietest bleven binnenstromen. Toen de teller op vijftig stond ben ik ze één voor één langsgelopen om te analyseren waarom de klant zoveel meer fouten vond dan ik. En dat in een veel kortere tijd. Ik was hier toch de gecertificeerde tester?

Wat bleek: Het gros van de meldingen had ik op basis van de specificaties niet kunnen vinden. Het waren afwijkingen ten opzichte van vorige versies of het was nieuwe functionaliteit die niet consistent was met bestaande functionaliteit. Voor een externe tester lastig te constateren, maar gesneden koek voor iemand die dagelijks met de applicatie werkt.

Welke conclusie kon ik hier nu uit trekken? In principe heeft het testplan gewerkt, want deze fouten had ik niet kunnen (of hoeven) vinden. Misschien is dat de juiste conclusie, menig tester zal dat zo zien. Maar ik was daar niet tevreden mee. Ik concludeerde dat om beter te testen je blijkbaar meer informatie nodig hebt dan alleen de ontwerpdocumentatie.

Exploratory testing als antwoord?

Het was ergens rond deze periode dat ik me begon te verdiepen in exploratory testing; een methode waarin het opstellen van testgevallen, testuitvoer en leren over een product hand in hand gaan.

Om deze methode te kunnen structureren maakt exploratory testing gebruik van korte testsessies, met een afgebakend doel. Eindresultaat van een sessie is een verslag met aantekeningen over opvallende zaken, bugs, uitgevoerde tests en screenshots.

Deze methode heb ik sindsdien bij meerdere opdrachten gebruikt. Door me niet te beperken tot de vooraf opgestelde testgevallen blijft het werk intellectueel uitdagend. Scripten op basis van een ontwerp is belangrijk, maar kan erg eentonig worden. Onderzoeken hoe duizend-en-een zaken die niet beschreven staan toch fout kunnen gaan is altijd verrassend. En minstens zo belangrijk.

Ik gebruik alle informatie die ik kan vinden over het product dat ik test.

Dit betekent ook vaak dat ik samen met een eindgebruiker test. Door de verslagen die ik maak, kan ik nog makkelijk terugvinden wat ik op bijvoorbeeld 11 mei 2010 heb getest, welke fouten ik die dag heb gevonden en wat ik over de applicatie heb geleerd. Voor mijn gevoel is het eindresultaat van mijn testen, nuttige informatie over de kwaliteit van de software, door deze methode met sprongen vooruitgegaan. Moet het eindresultaat van testen enkel een metriek zijn? 95 testgevallen uitgevoerd, 92 akkoord, 3 niet akkoord waarvan één stopper? Ik denk dat een testverslag betere informatie kan bevatten, maar daarvoor moet je een applicatie wel hebben onderzocht. Is dit dan het verhaal waar mijn manager naar op zoek is? Is exploratory testing dan de methode waarmee je je als tester kan onderscheiden? Deels. Soms. →



Het belang van context

Bij mijn huidige opdracht lopen de bouw van nieuwe functionaliteit en het oplossen van klantmeldingen door elkaar heen. Voor het testen van nieuwbouw heb ik exploratory testing op basis van testsessies geïntroduceerd. Dit werkt tot nu toe naar grote tevredenheid. Het hertesten van opgeloste productiemeldingen vereist juist weer de lessen uit TMap over het inrichten van een testorganisatie: inrichten en beheer van testomgevingen, de risicoanalyse, het opstellen van checklists, het maken van afspraken over rapportage en overdracht.

Er zal ook nieuwe functionaliteit aankomen waarvoor testen met ketenpartners moeten worden uitgevoerd. Hiervoor helpt kennis over TMap of exploratory testing niet. Hiervoor moet je iets weten over de branche. In mijn huidige opdracht betekent dat, dat je iets moet weten over de zorg in Nederland. Welke zorginstelling declareert wat waar? Hoe leidt een indicatiestelling tot een zorgtoewijzing? Wat is een MAZ en een MUT? Dit soort zaken weet je niet vooraf, die ontdek je pas als je bij de klant aan het werk gaat en inventariseert waar op testgebied behoefte aan is.

En juist daar ligt de toegevoegde waarde van de tester. Niet op een eerste afspraak gelijk aan een klant vertellen hoe er getest moet worden. Dé juiste testmethode bestaat immers niet. Welke methode je kiest, is afhankelijk van de context bij de klant. En om die context inzichtelijk te maken zullen veel vragen beantwoord moeten worden:

- Hoe volledig zijn de ontwerpen?
- Moeten testgevallen overgedragen worden?
- Hoeveel capaciteit is er voor testen beschikbaar?
- Wie beschikt in de organisatie over welke kennis?
- Is het een lijn- of projectorganisatie?
- Wordt via Scrum of de watervalmethode ontwikkeld?
- In welke mate test development haar eigen werk?
- Wie zijn de stakeholders voor een go / no-go beslissing?
- Is de opgeleverde software regressiegevoelig? Op welke onderdelen?
- Beschikt de organisatie over styleguides?
- Aan welke SLA's is de organisatie gebonden?
- Welke tooling kan voor het testen worden ingezet?
- Blijft de opgeleverde software relatief ongewijzigd, of wordt deze continu aangepast?
- Wat zijn verwachtingen van de organisatie over de output van testen?
- En zo nog vele vragen...

Herkennen waar de klant behoefte aan heeft, je daarbij niet beperken tot één methodiek, maar onderzoeken en aanbieden wat de klant nodig heeft, dat is het verhaal. Mijn verhaal in ieder geval en dat is nog lang niet af.

Maarten: 'Zou je dat allemaal eens op papier willen zetten?'

Stefan: 'Is goed, zal ik doen!'

We zijn halverwege ons vierde biertje.

Stefan: 'Zo, genoeg over werk. Die Balotelli, wat een figuur is dat, zeg...' →

Conclusie

Voor salesmanagers is mijn antwoord geen prettig antwoord. Een standaard verhaal over goed testen bestaat niet. De testbehoefte zal per klant verschillen en er zijn vele vragen te stellen voordat je een realistisch beeld hebt over hoe er het best binnen een organisatie getest kan worden. Een beeld dat waarschijnlijk ook regelmatig bijgesteld zal moeten worden.

Zijn wij als testers wel flexibel genoeg om ons aan verschillende organisaties aan te passen? Willen wij verder kijken dan een functioneel ontwerp om te onderzoeken hoe een applicatie werkt? Accepteren wij de grote mate van onzekerheid die er bij testen komt kijken? Wat doen wij als testers om beter te worden in ons vak? Om betere informatie over software kwaliteit te kunnen verstrekken? Het zijn geen makkelijke vragen. Testen is dan ook geen makkelijk vak.

Suggesties voor verder lezen:

- Exploratory testing: <http://www.satisfice.com/articles/et-article.pdf>
- Session-based test management: <http://www.satisfice.com/articles/sbtm.pdf>
- Context-driven testing: <http://context-driven-testing.com/> ←

EUROSTAR 2012 VOLGENS STÉPHANIE

Door Stéphanie Heidstra • heidstra@gmail.com



Moe maar voldaan loop ik, op de Expo van EuroSTAR, Rob van Steenberg tegen het lijf. Ik ben al vier dagen terloops in gesprek met bekende en minder bekende gezichten vanuit TestNet, dus het is geen opzienbarende ontmoeting. Terloops komt ter sprake dat er nog een stukje geschreven moet worden voor TestNetNieuws en in een vlaag van verstandsverbijstering hoor ik mezelf zeggen dat ik dat 'wel even doe'.

Rob kijkt blij verrast en noteert meteen mijn e-mailadres. Direct daarna ontsnapt Rob, ik hoor hem nog wat zeggen over hoe moeilijk het is om mensen te krijgen voor het schrijven van stukjes. En zo komen we hier terecht. Op zondagochtend, met een kopje koffie achter mijn bureau op zolder. En nu eerst even een stukje schrijven voor TestNetNieuws. Het zijn vier drukke dagen geweest en na alle gesprekken en presentaties heb ik een flinke klus aan het ordenen van alle nieuwe informatie en ideeën. Daarnaast heb ik natuurlijk ook dat stukje nog om te schrijven. Eerst moet ik verzinnen welke vorm het krijgt. Verslagen in de eerste persoon, die vaak een opsomming worden van: 'en toen ... en toen ... en toen ...', die lees ik niet graag. Dan ben ik al snel de draad kwijt en verlies ik de interesse, dus een dergelijke opzet wordt het niet.



Misschien kan ik EuroSTAR uitdrukken in getallen? Dan krijg je een stukje dat er ongeveer zo uit ziet:

- vier dagen EuroSTAR;
- vijf hele dag tutorials, vijf halve dag tutorials, vijf keynotes;
- vier lunches, acht koffiepauzes (met koekje), één galadiner;
- vier actieve workshops;
- veertig presentaties;
- achtenvijftig stands in de Expo.

Maar erg interessant ziet dat er ook niet uit. Misschien dat het helpt als ik het wat persoonlijker maak? Thuis aangekomen blij ik verzameld te hebben:

- elf foldertjes van tools die nader bekeken moeten worden;
- veertien T-shirts en acht EuroSTAR tassen;
- dertig pennen;
- vier sessie-readers en werkboeken;
- tien visitekaartjes;
- drie oefenexamens ISTQB en gratis toegang tot een online bibliotheek;
- minimaal zes maanden werk om alle nieuwe ideeën uit te werken en een plaats te geven;
- één uitnodiging voor een sessie over de nieuwe ISO-standaard bij Valori;
- één sticker. →



Al met al blijft het droge kost, bovendien is het wel erg feitelijk. Rikard Edgren zou het zeker zien als een symptoom van de binaire ziekte, die hij bij veel testers gediagnostiseerd heeft. Hij pleit voor een meer subjectieve rapportage en het loslaten van de nu veel gebruikte methode om te rapporteren over Pass/Fail en aantallen gerunde tests. Ik heb zo mijn twijfels over zijn verhaal en denk dat een rapportage altijd een objectieve onderbouwing moet hebben, maar laat ik niet te veel te koop lopen met het feit dat ik zeker besmet ben en dat zijn medicijn nog niet tot genezing geleid heeft.

Misschien dat het interessanter wordt als ik het ophang aan een alfabet

Het EuroSTAR alfabet zou dan een stukje opleveren zoals:

A

De A staat voor Automatisering van tests. Daar waren veel en heel goede tracks over te vinden. De dag-sessie van Dorothy Graham gaf veel praktische pointers voor testers die hun bestaande test willen verbeteren. Ook had zij veel metrics die iets kunnen zeggen over de kwaliteit van een geautomatiseerde test. Dorothy heeft een interessant nieuw boek, waar je heel EuroSTAR niet omheen kon, en ze stelt op haar site de nodige tools en informatie beschikbaar (<http://www.dorothygraham.co.uk/>)

B

De B staat voor Binary ziekte, waar veel testers gevoelig voor blijken te zijn.

C

De C staat voor Community Hub, waar veel vijfminutenpraatjes discussies uitlokten, over de goede toepassing van testproces verbetering bijvoorbeeld.

D

De D staat voor Defects, die kun je bemonsteren om er nog meer te vinden. Randall Rice promoot een aantal methoden om meer defects te vinden. Deze methoden zijn gestoeld op het feit dat defects clusteren: ze zitten vaak in de code bij elkaar. Door een gedegen defectanalyse kun je dus meer fouten vinden omdat je je test kunt focussen op gedeeltes software waar aantoonbaar al fouten zitten. Als je deze technieken toepast in het test-lab bij EuroSTAR, dan krijg je een T-shirt!

E

De E staat natuurlijk voor Excellence-award, die zo vakkundig in de wacht gesleept is door Bob van de Burgt. Of voor Expo, wat jij wilt. →

G

De G staat voor Gangnam-style: die heb je pas goed gezien als Bart Knaack, Bob van de Burgt en Graham Freeburn hem uitvoeren. Filmpje: <http://www.youtube.com/watch?v=jcM4Cb8-ueg>



H

De H staat voor Heisenbugs. Dit zijn de meest ongrijpbare bugs die veroorzaakt worden door race-conditions of timingproblemen als software uitgevoerd wordt op meer dan één core. Mike Bartley had hier een verhaal over, maar dat verhaal riep meer vragen op dan dat het antwoorden opleverde. Het is wel duidelijk dat software die niet lineair verwerkt meer eisen stelt aan multi-user tests.

I

I staat voor iPad mini en X staat voor X-Box. Het was op de Expo mogelijk om geen van beide te winnen.

M

De M staat voor Mindmaps, die je werkelijk overal voor blijkt te kunnen gebruiken, het schrijven van een stukje bijvoorbeeld. Maar Fiona Charles had een handige aanpak om met een mindmap tot een teststrategie te komen. In een notendop begint ze vanuit het project met het identificeren van de stakeholders. Vervolgens bepaalt ze per stakeholder welke waarde het project voor ze heeft: wat hebben ze te winnen en wat vinden ze belangrijk aan het project? Hierna is het mogelijk om per waarde risico's te identificeren en beperkingen vast te leggen. Deze laatste leiden eigenlijk vanzelf naar een teststrategie.

S

De S staat voor Spokenjacht wat, volgens Alan Richardson een prima invloed op het testen kan hebben. Zijn praatje was wel inspirerend, maar bracht weinig direct praktisch toepasbare nieuwe inzichten.

T

De T staat voor ondergewaardeerde Tools, Graham Thomas had hier een verhaaltje over vlak na de lunch, gelukkig was het niet zo nodig om je hoofd erbij te houden. →

Z

De Z staat voor Zeven vragen, waarmee Jean-Paul Varwijk je goed op weg helpt.

En de V die staat voor Veel te veel letters die ik overhoud met deze aanpak. Dit leidt dus ook tot niets.

Toch zou er genoeg te schrijven moeten zijn over EuroSTAR. Het zijn vier actieve dagen, waarin voldoende kennis en ervaring gedeeld wordt voor een heel jaar. Er hangt een goede sfeer en mensen leggen actief contacten met elkaar. Bijna iedereen loopt wel een vorm van inspiratie op.

Er lijkt niets anders op te zitten dan Rob maar even mailen: sorry Rob, een stukje schrijven over EuroSTAR, het lijkt me niet te lukken, misschien volgend jaar beter. ←

MAPPA TESTI

Door Nathalie Rooseboom de Vries • funtestic.fanatic@gmail.com • [@funTESTic](https://twitter.com/funTESTic)



Wie het jubileumboek van dit jaar van TestNet heeft gelezen, heeft in ieder geval al een versie van een Mappa Testi gezien; zowel in het beeldscherm van de iPhone op de voorkant en op de uitvouwbare kaft van het boekje staat een speciaal getekende editie van de kaart afgebeeld. Maar wat is de oorsprong van de originele Mappa Testi, waar dient hij voor en hoe maak je er zelf één?

Het ontstaan van de originele Mappa Testi

Het begon allemaal tijdens de Testers Retreat in Hereford in januari 2012. De testers retreat is een groep van twaalf à veertien testers die een huis huren voor een weekeinde om te discussiëren over testen in al haar facetten. Tijdens het weekeinde wordt er ook een ontspanningsmoment ingelast en deze editie was dat een bezoek aan de oudste, nog bestaande middeleeuwse wereldkaart in de kathedraal van Hereford; de Mappa Mundi.



De BBC heeft een heel mooie serie gemaakt over de Mappa Mundi die je op YouTube kunt vinden. In de filmpjes (totaal vier stuks die één geheel vormen) wordt uitgelegd hoe de kaart in elkaar steekt. De kaart is namelijk niet alleen een topografische wereldkaart (overigens zonder Amerika en Australië, want die kende men toen nog niet), maar is een beeld dat men toentertijd had van religie, toont fauna die men kende, fantasiedieren,, fabels en mythen en toont tevens een reis door de tijd evenals een plaatsing in de context van het leven.

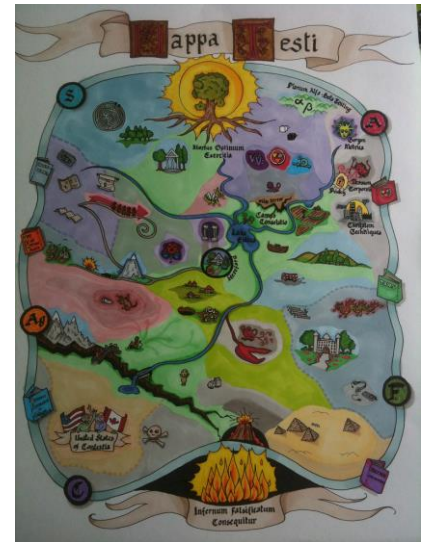
Het middelpunt van de Mappa Mundi is Jeruzalem en op de kaart is ook heel duidelijk te zien dat men bepaalde delen van de kaart veel nauwkeuriger heeft ingetekend (dat kende men) en dat men bepaalde delen invult met zaken die men kent 'van horen zeggen'. Zo is de rode zee daadwerkelijk rood gekleurd. Men had die zee nog nooit gezien, maar kleurde hem dus rood vanwege zijn naam. Dieren die in Europa zijn getekend zijn nog redelijk herkenbaar, maar hoe verder men komt, hoe fantasierijker de wezens worden ... De Mappa Mundi is een feest om te ontdekken.

Na het bezoek aan de Mappa Mundi keerden we geïnspireerd terug bij het huis van de retreat. We besloten een Mappa Testi te maken, waar we de testing wereld in konden tekenen zoals we deze als retreaters zagen. We startten met het middelpunt; Hereford. Dit is immers de plaats waarvandaan we de kaart tekenen. Vervolgens tekenden we een grote grens; in de echte Mappa wordt de wereld verankerd op de kaart door de letters 'MORS', oftewel men wordt begrensd op de wereld door de dood. Onze interpretaties hiervan werden de 'testprincipes': 'Agile', 'Factory', 'Standards', 'Automation' en 'Context'. →

Na het bezoek aan de Mappa Mundi keerden we geïnspireerd terug bij het huis van de retreat. We besloten een Mappa Testi te maken, waar we de testing wereld in konden tekenen zoals we deze als retreaters zagen. We startten met het middelpunt; Hereford. Dit is immers de plaats waarvandaan we de kaart tekenen. Vervolgens tekenden we een grote grens; in de echte Mappa wordt de wereld verankerd op de kaart door de letters 'MORS', oftewel men wordt begrensd op de wereld door de dood. Onze interpretaties hiervan werden de 'testprincipes': 'Agile', 'Factory', 'Standards', 'Automation' en 'Context'. →

Hierna tekenden we onze 'hel'. Als groep waren we het er over eens dat 'falsified results' echt niet kunnen in het beroep van tester. Hiermee gingen we ook afwijken van hoe de werkelijke Mappa is opgezet. Op de Mappa Mundi zijn de niet-aardse zaken als 'hel' en 'hemel' buiten de kaartgrenzen getekend (bovenaan wordt de 'dag des oordeels' afgebeeld). Wij vonden het leuker om de 'hel' op de bodem van de kaart te tekenen en de 'hemel' bovenaan. We bedachten dat veel testers 'Best practices' als 'hemel' zien; we tekenden hem als de 'Garden of Best Practices' (geïnspireerd op de tuin van Eden). De achterliggende gedachte hierachter is dat je zelf kunt bepalen of je gelooft in de tuin van Eden uiteraard ☺

Bij de Mappa Testi zijn we vervolgens de dingen gaan tekenen die ons (willekeurig?) te binnen schoten waarvan we vonden dat ze op de kaart zouden moeten staan. We groepeerden die zaken enigszins, waarbij we de voor ons heel bekende dingen dicht bij het centrum hielden en de dingen die we niet fijn vonden of die minder bekend waren verder van ons centrum af tekenden (zoals dat ook op de Mappa Mundi wordt afgebeeld). Zo vonden we allemaal 'ethiek' heel belangrijk en hebben we het meer van Ethiek dicht bij 'Hereford' getekend. Zo waren de meesten van ons minder bekend met 'Context driven' of hadden we daar een vooroordeel over, wat we hebben vertaald naar 'de United States of Contextia' die worden gescheiden door een grote kloof van onbegrip en waar een groep van Agilista's een brug over probeert te maken.



Hoe de map precies is opgebouwd kun je zien op het volgende YouTube filmpje: <https://www.youtube.com/watch?v=Vrfzl8Z9nOs> welke het proces van tekenen en mijn uitleg van de Mappa weergeeft. Derk-Jan de Grood maakte het filmpje.

Leuk zo'n plaatje, maar wat heb ik er aan?

Net als de Mappa Mundi heeft de Mappa Testi verschillende lagen van betekenis en kan het maken van een eigen Mappa een heel creatieve manier zijn om elkaar (en elkaars visie op de testwereld) te leren kennen.



Tijdens het tekenen van de originele Mappa Testi had de groep veel discussie over de onderwerpen die op de kaart getekend werden. Iemand riep een term en vervolgens kwam er discussie over de plaats waar die getekend moest worden en de onderbouwing waarom. Dat had tot gevolg dat er uitleg kwam over de context en de visie van die persoon op het onderwerp. Diegenen die weer minder van het onderwerp af wisten kregen op die manier weer meer informatie en kennis. Ook de wijze waarop iets getekend moest worden had discussie tot gevolg; teken je iets als monster of juist als iets lieflijks? Het had te maken met de mate waarin iemand met het onderwerp een affiniteit had of juist niet. Als iedereen het gelijk eens was over een te tekenen onderwerp leidden we daaruit weer af dat er misschien iets moest veranderen in de testwereld of dat iets juist meer belicht mag worden. Eén ding was zeker; de Mappa had veel begripsvorming en discussie tot gevolg.

Het maken van een eigen Mappa Testi, zoals een Mappa van je project waar je mee bezig bent, kan inzicht geven in de gebieden die onbelicht zijn of waar je je juist comfortabel mee voelt. Waar je naar toe wilt bewegen (wandelpaden en pootjes) of wat je juist verschrikkelijk vindt (hel). →

Een Mappa zal niet direct een offensief stuk zijn waar mensen zich beledigd door kunnen voelen, maar is juist een vriendelijke manier en een discussiestuk dat tot vragen inspireert. Iemand die de werkvloer oploopt en een Mappa

ziet zal sneller geneigd zijn om te vragen 'Wat is dat?'. En jij zult sneller geneigd zijn om het uit te leggen wat de dingen betekenen dan dat een groot bruin vel doet met post-it's waarop staat wat nog mist en wat goed gaat.

Een Mappa Testi kan voor ieder 'niveau' getekend worden of voor ieder doeleinde. De Mappa Testi voor het jubileumboek is hier een goed voorbeeld van. De 'landen' zijn de hoofdstukken, de 'principes' zijn de schrijvers van het boek, het centrum is 'TestNet'. Dat is het startpunt of de inspiratie waarvandaan het boek ontstaan is.



Ik heb ook een persoonlijke Mappa gemaakt. Tijdens EuroSTAR, afgelopen november, had ik deze Mappa ook tentoongesteld. Mensen vroegen me naar de betekenis en ik legde hen het uit. Binnen twee minuten kon men zich een beeld vormen wat voor tester ik was. Iemand die van formeel en standaardisering houdt, maar ook graag kennis deelt en waar mijn persoonlijk leven een grote inspiratie is voor mijn werk als software tester.

Tijdens de workshop die ik op donderdagmiddag op EuroSTAR hield tekenden mensen ook hun eigen Mappa. Aan het einde van de workshop was er een veelvoud van Mappa's (Mappa-galerie) waar in een zeer korte tijd iedereen elkaar leerde kennen op een ongedwongen manier en creativiteit – ongeacht de tekenkwaliteiten – tot mooie verhalen van de deelnemers leidden. De tekeningen varieerden van hun opgedane ervaringen op EuroSTAR tot persoonlijkere mappa's met de 'landmarks' van ervaringen die de persoon hadden gevormd tot de tester die ze nu waren. En 'monsters' of 'onontdekte gebieden' waar anderen dan op reageerden met: 'ik kan je die helpen ontdekken!'.

Je kunt Mappa's tekenen voor 'project', als vervanger van een gewone mindmap, van 'bedrijf', 'persoon' of zelfs 'vakgebied'; zo zou ik bijvoorbeeld erg graag de Mappa zien van een 'Context Driven' groep; hoe zien zij eigenlijk de testwereld?

Je eigen mappa maken

Belangrijk bij het maken van je eigen Mappa is dat je je niet laat afschrikken dat het een getekend iets is. Je kunt ook stickers of geknipte plaatjes gebruiken. De manier waarop je een mappa maakt is niet belangrijk, het is de uiteindelijke kaart waar het om gaat. Wel belangrijk is dat je er geen 'stroomdiagram' van maakt. Het gaat juist ook een stukje creatief bezig zijn (proces dat de mappa tot gevolg heeft) en het kunstzinnige dat aanzet tot de vragen: krullige, niet perfecte lijnen zetten aan tot een nader onderzoek ... Een geheel met de computer gemaakte afbeelding zal niet 'triggeren' tot vragen, maar zal worden gezien als een zoveelste onbeduidende afbeelding van een computersysteem op de muur en dan mis je het hele punt van inspiratie. →

Een Mappa is je *eigen* (of groepsvisie) van iets. Anderen kunnen het er helemaal niet mee eens zijn. Mijn antwoord: laat ze een eigen Mappa tekenen, die hang je eraan en dan heb je een mooi beeld dat meningen kunnen verschillen.

Een *mogelijk* stappenplan:

- Teken een titelbalk;
- Teken het centrum en de grens (die grens kun je ook als laatste tekenen als je al je onderwerpen ingetekend hebt om te voorkomen dat je ruimte te kort komt ☺);
- Teken wat je het meest verschrikkelijk vindt onderaan (hel);
- Teken wat je het meest prettig vindt bovenaan (hemel);
- Teken de dingen die je het meest naast liggen dicht bij het centrum;
- Teken de dingen die je onbekend zijn verder van je centrum weg.
- Teken de dingen die je niet leuk vindt met een monsterachtig uiterlijk;
- Teken de dingen die je wel leuk vindt met een lieflijk uiterlijk;
- Wat zijn de principes die je aan 'deze wereld' binden; die komen op de grens;
- Denk aan: wat zijn historische 'markers' in je loopbaan of project, wat vormde jou als tester, wat zijn de onbekende dingen, wat wil je leren, wat kun je andere mensen leren, wat zijn dingen die je in de toekomst ziet? Wat is je inspiratie enzovoort;
- Pseudo Latijn maakt het mysterieuzer en zet aan tot vragen, maar alles is goed. Een woord met 'us' erachter maakt het al apart ☺, dan klinkt 'boundary analysis' als 'boundarus analyticus' bijvoorbeeld;
- Pijltjes, voetstappen en lijnen geven aan dat je ergens naar toe wilt of juist vandaan wilt;
- Details maken het geheel interessanter net als het gebruik van veel kleuren;
- Je kunt post-its gebruiken om je positionering eerst te doen, vooral als je het 'eng' vindt om direct te tekenen kan dit helpen;
- Je kunt in plaats van te tekenen ook afbeeldingen gebruiken vanaf internet en/ of PowerPoint, Visio enzovoort. Maar let op: je kunt die plaatjes dan het beste uitprinten en op je map plakken. Direct een map op de computer maken, maakt het geheel te klinisch en zet niet aan tot nieuwsgierigheid.



Meer weten van de Mappa Testi of hoe je een Mappa-workshop op kunt zetten voor jou en je team? Neem dan gerust contact op! Mijn e-mailadres is: funtestic.fanatic@gmail.com ←

TESTEN IS KINDERSPEL

Door René Tuinhout • ReneTuinhout@ismijnaam.nl



Laatst was ik op bezoek bij kennissen. Sinds enige tijd hebben zij een zoontje, Jochem, en het kereltje is nu bijna een jaar oud. En Jochem is overduidelijk in de testfase beland: In zijn kinderstoel gezeten bekijkt hij zijn nog kleine wereld. Wat voor Jochem interessant is, is de wereld op het tafeltje voor hem. Waarop altijd wel een speelgoedje of een tuimelbeker staat. Heeft Jochem de tuimelbeker ontdekt, dan plaatst hij zijn rechterhand links van de

beker en beweegt die hand langzaam, tergend langzaam, naar rechts. Even stopt zijn beweging als de rand van het tafelblaadje bereikt wordt, maar vervolgens duwt hij met een frons in zijn voorhoofd ferm door, tot de beker over de rand van de tafel uit het zicht verdwijnt.



Binnen een seconde of vijf klinkt er dan een gillette, gevolgd door een 'Jochem toch, wat doe je nu', waarna weer een seconde of tien later, na wat stampende geluiden, de beker op mysterieuze wijze weer op de kinderstoelafel verschijnt.

Onderzoekend

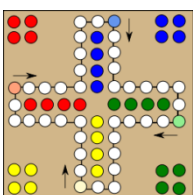
Nu is Jochem een onderzoekend kind. Vlak voor het weer verschijnen van de beker op zijn tafeltje kun je Jochem met verbazing naar die beker zien kijken en verschijnt weer die frons. Je ziet hem haast denken: 'zouden duwen-uit zicht verdwijnen-gillette-stampen-herverschijnen acties zijn die met elkaar verbonden zijn'? Nu is Jochem een onderzoekend baasje. Dus wat doet Jochem? Jochem plaatst zijn rechterhand links van de beker en beweegt die hand langzaam, tergend langzaam, naar rechts. Weer stopt zijn beweging even als de rand van het tafelblaadje opnieuw bereikt wordt, de frons verschijnt weer, en weer verdwijnt de beker over de rand van de tafel uit het zicht.

En weer klinkt een gillette, hoort Jochem het stampen en verschijnt de beker plotseling weer voor hem op tafel. Jochem is zo gebiologeerd door dit proces dat hij maar niet moe wordt dit te herhalen. Zijn moeder staat zo'n vijftien keer op van de bank om zijn beker op te rapen tot ze uit arren moede de beker maar in Jochems mond duwt, hem vasthoudt, en wacht tot de beker leeg is.

En Jochem? Jochem leert een heleboel: Er is inderdaad een verband tussen duwen en gillette's! En dingen die je over een tafeland duwt verdwijnen (als de beker is verwijderd zal Jochem dit toetsen met zo ongeveer al het speelgoed op zijn kinderstoelafeltje)!

En zo hebben we ooit allemaal in een kinderstoel gezeten, en ontdekten we de zwaartekracht. En de onuitputtelijkheid van volwassenen als het terugbezorgen van je gevallen tuimelbekers, knuffels en blokken betrof.

Geleerd



Jochem gaat opgroeien. En zijn tijd in de kinderstoel vergeten. Maar wat hij niet vergeet, is wat hij er geleerd heeft: Dingen die je over de rand van een tafelblad duwt verdwijnen (hij weet later zelfs dat ze dan vallen). Tegen de tijd dat Jochem een jaar of zeven is, kan hij al aardig meedoen met spelletjes. En ook dan gebeurt er iets interessants: Jochem speelde ook op jongere →

leeftijd spelletjes, maar rond zijn zevende gaat hij ontdekken dat hij vals kan spelen. Heel opzichtig, door veel misbaar te maken als hij geen zes gooit bij Mens erger je niet. In de hoop dat veel tranen, zielig kijken en decibels richting zijn vader of moeder, hen kunnen doen besluiten hem nog een poging op zes gooien te gunnen!

Maar ook heel subtiel, door bij kwartet te gaan vragen naar kaartjes uit een serie waarvan hij zelf geen kaartje bezit. En dan weer stiekem kijken naar z'n ouders of die dat niet doorhebben. Waar Jochem op zijn eerste leerde dat je een verwachting kunt toetsen, leert hij rond zijn zevende dat je uitkomsten kunt beïnvloeden.

Vooraf toetsen

En Jochem zal nog meer gaan leren! Tegen de tijd dat hij zo'n jaar of twaalf, dertien is en de puberteit hard toeslaat, zijn Jochems leeftijdsgenoten opeens belangrijk voor hem. Zijn kleding valt binnen de normen van z'n groep, z'n gedrag, of hij wel of niet rookt, welke sport door de beugel kan en welke niet, bijna alles moet binnen normen vallen. Niet door maar iets te proberen en te kijken wat er gebeurt, zoals Jochem rond z'n eerste met tuimelbekers probeerde. Ook niet door achteraf te proberen om een resultaat te beïnvloeden. Nee, in de puberteit zal Jochem gaan proberen om vóóraf te toetsen: Die broek die hij mooi vindt, wat vinden zijn leeftijdsgenoten daarvan? Dat gaat Jochem uitzoeken vóór hij zijn schaarse kleedgeld uitgeeft. Wat is de mening van de populaire jongens uit zijn groep over een bepaalde sport? Dan is dat ook Jochem's mening. Misschien met wat kleine eigen nuances, maar hij toetst of zijn mening door de beugel kan, voordat hij hem geeft.



En in het leerproces dat Jochem gaat doorlopen zit iets heel bekends:

- verwachtingen vormen en daar tegen testen, lijkt dat niet erg op testgevallen maken?
- verwachtingen proberen te beïnvloeden en daar tegen testen, lijkt dat niet erg op extra informatie inwinnen en op basis daarvan testgevallen maken?
- vooraf nagaan wat door de beugel kan, en daar een mening over hebben? Lijkt dat niet erg op toetsen, statisch testen, reviews en inspecties?

En zo zien we: we zijn allemaal als testers geboren, we groeien op als testers, we worden volwassen als testers. Velen hebben dat niet door (en maken dan een foute beroepskeuze en worden bouwer of zo ☺), maar enkele slimme geesten besluiten hun leven als tester voort te zetten.

Wat kunnen wij als professionele tester van dit kinderspel leren?

Bij de eenjarige kunnen we denk ik bewondering hebben voor hun volharding: test en test en test nog een keer, net zolang tot het goed is, of totdat hogerhand besluit dat het niet meer getest hoeft te worden.

Van de zevenjarige kunnen we leren dat het verstandig is om uit te vinden hoe de hazen lopen, wie het voor het zeggen heeft en daarop in te spelen. Als er geen draagvlak meer is, heeft verder spelen geen zin.

En van de puber kunnen we leren dat vroegtijdig toetsen een hoop missers kan voorkómen.

Het thema van deze TestNetNieuws vind ik perfect: Testen is inderdaad kinderspel! →

Laten we daarom het kind in onszelf niet vergeten en met volharding, inzicht en vooruitkijken onze tests blijven uitvoeren.

Oh, nog even over Jochem. Jochem weet van dit alles nog niets. Hij is één, zit in zijn kinderstoel en gooit zijn beker op de grond. Gezien zijn fascinatie voor het herhalen van tests: Is er nog iemand op zoek naar een gefocuste tester voor regressietesten? ←

CREATIVITEIT BIJ NAUWKEURIGE TESTEN VAN GELUID BIJ TELEFONIE

Door Peter Roozendaal • pf.roozendaal@gmail.com



Voor een VOIP telefoniesysteem werd als eis gesteld dat de end-to-end vertraging in het spraaksignaal niet meer dan 150 msec mag bedragen. Dit artikel beschrijft een methode om die vertraging eenvoudig, nauwkeurig en goedkoop te meten.

In een ander geval bleek een te testen Voice Response Systeem de ingevoerde geluidsfragmenten te verminken. Met dezelfde tool was eenvoudig te analyseren waar het mis ging.

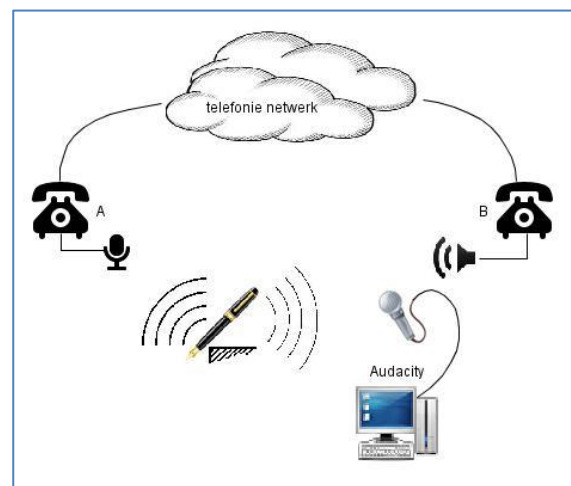
Vertragingen meten in een VOIP telefoniesysteem

Om een prettig verlopend telefoongesprek mogelijk te maken is het belangrijk dat de vertraging in het spraaksignaal niet te groot is. De eis die vaak gesteld wordt is dat het signaal van telefoon A maximaal 150 msec later aankomt bij telefoon B. Bij 'normale' telefonie (PSTN, ISDN) geeft dat geen problemen, maar bij VOIP (Voice over IP) telefonie ligt dat anders. De packetsize van de voice pakketjes (normaal 10 msec), de grootte van de jitter buffer, instellingen van firewalls en routers en het aantal VOIP-ISDN conversies in de verbinding maken het soms lastig aan de 150 msec eis te voldoen.

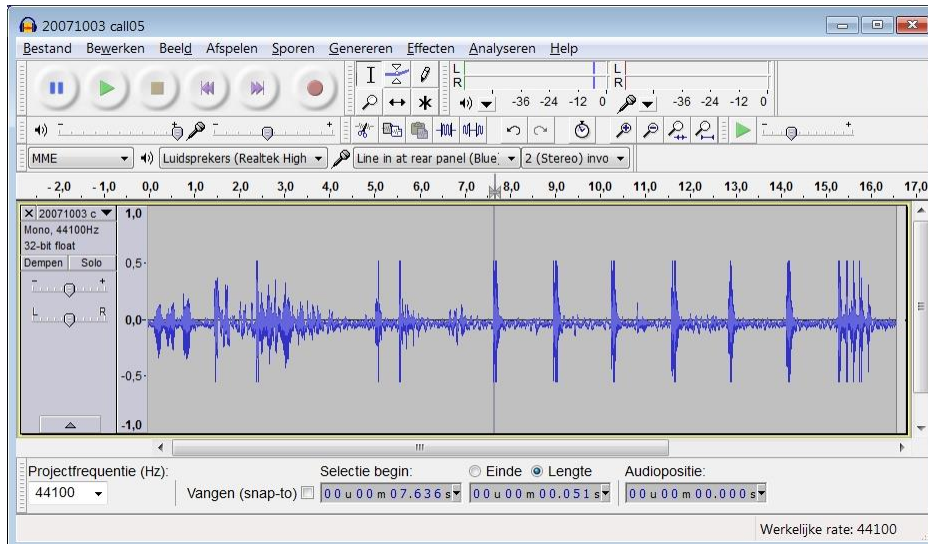
In een testopstelling met een nieuw VOIP telefoniesysteem waren volop telefoons aanwezig waarmee allerlei verbindingen opgebouwd konden worden, al dan niet via het te testen VOIP systeem. Wanneer ik de hoorns van twee telefoons links en rechts tegen mijn oren hield, kon ik bij een ISDN-verbinding nauwelijks een vertraging horen, maar bij een VOIP-verbinding wel. Dat bracht me op het idee een puls vormig geluid (tik met pen tegen tafelrand) te maken dat door telefoon A naar telefoon B gaat waar het door een microfoon wordt opgenomen. Omdat alles naast elkaar staat wordt ook de originele puls door de microfoon waargenomen.

De microfoon is op een laptop of pc aangesloten. Voor het programma dat het geheel opneemt gebruik ik Audacity (zie figuur 1). Bij afwezigheid van een losse microfoon is het ook mogelijk de ingebouwde laptopmicrofoon te gebruiken, maar dan is een tweede tester nodig om alles op zijn plek te houden.

Het opgenomen signaal ziet eruit als in figuur 2. In de praktijk is het handig om even het testnummer in te spreken en eventueel wat details over de test. Door meerdere tikken op te nemen zit er altijd wel een tussen die mooi overkomt. →

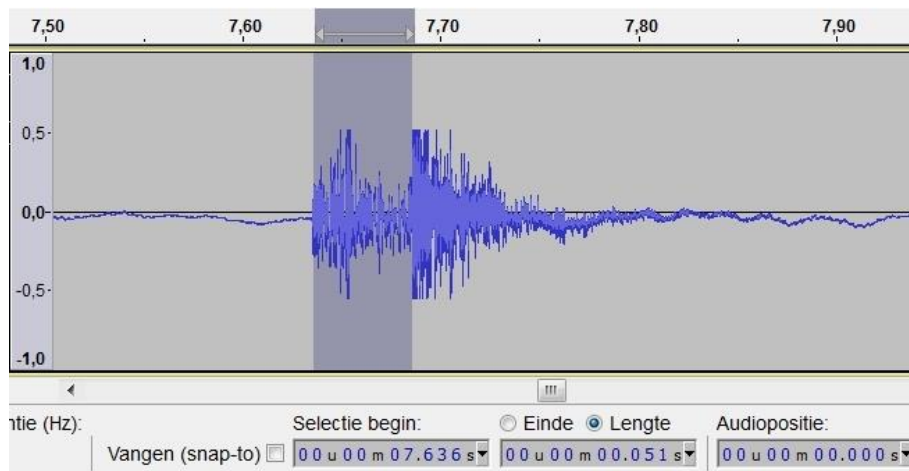


Figuur 1



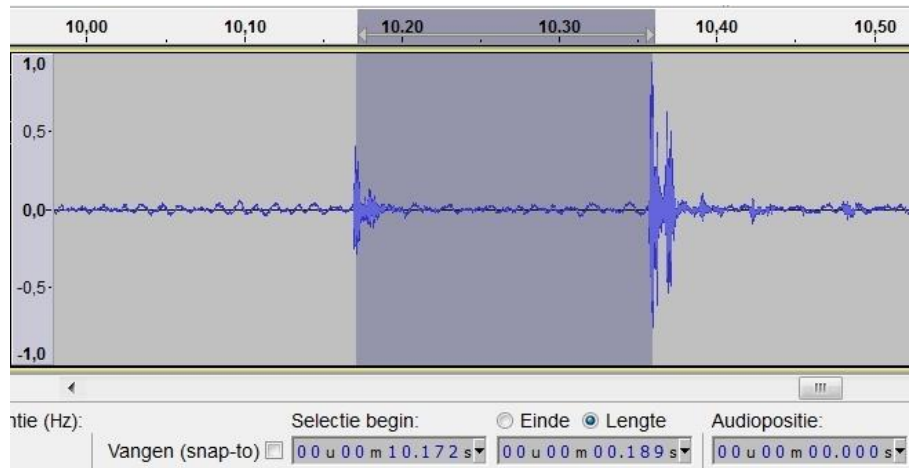
Figuur 2

Ingezoomd op één van de tikken is het op een eenvoudige manier mogelijk het signaal te selecteren vanaf het begin van de tik zelf tot de in B ontvangen tik, zie figuur 3. Hierna is af te lezen dat het geselecteerde deel 51 msec lang is. Een acceptabele waarde voor de hier opgenomen verbinding zonder VOIP component er in.



Figuur 3

Met een VOIP-verbinding werd het resultaat duidelijk anders, zie figuur4. Met een vertraging van 189 msec lag hier nog een uitdaging voor de leverancier. →



Figuur 4

Het gebruikte programma Audacity® is gratis, opensourcesoftware voor het opnemen en bewerken van geluiden is te downloaden op <http://audacity.sourceforge.net/>.

Behalve voor het opnemen en meten aan geluid kan Audacity ook gebruikt worden voor bijvoorbeeld het genereren van geluid, zoals gebruikt is in de volgende situatie.

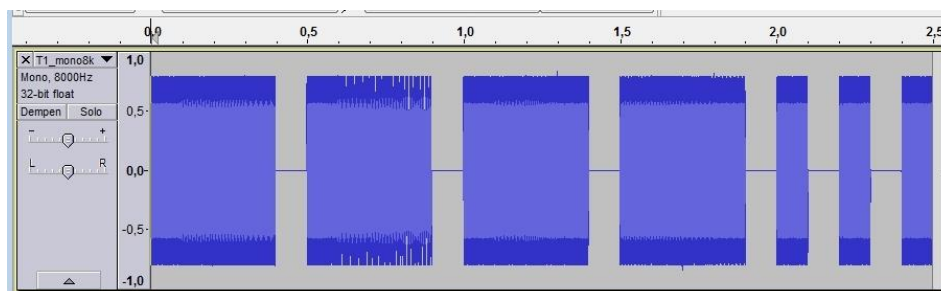
Missend stukje geluid opzoeken

Bij tests aan een telefonie Voice Respons Systeem ('voor inlichtingen toets 1, voor ...') bleek dat bij migratie van de voice-prompts van het oude naar het nieuwe systeem vaak een stukje van het begin ontbrak. Omdat voice-prompt vaak beginnen met een korte stilte was ook niet zeker of alle prompts aangetast werden. Het verdwijnen van een stukje stilte is dan onhoorbaar. Het was niet duidelijk of dit kwam door de extractie uit het oude systeem, bij het aanbrengen in het nieuwe systeem, of bij afspelen zelf.



Figuur 5

Met de toongenerator functie van Audacity (figuur 5) heb ik een geluidfragment gemaakt dat bestaat uit een soort kort melodietje van 100 msec toontjes en stiltes (zie figuur 6), en wel zodanig dat hoorbaar en zichtbaar zou zijn wat er na aanbrengen en afspelen mee gebeurde. →



Figuur 6

In figuur7 is te zien hoe de 100 msec fragmentjes elkaar opvolgen. Door stukjes toonladder te maken, afgewisseld met 100 msec stiltes bleek ook met het oor goed te bepalen wat er misging.

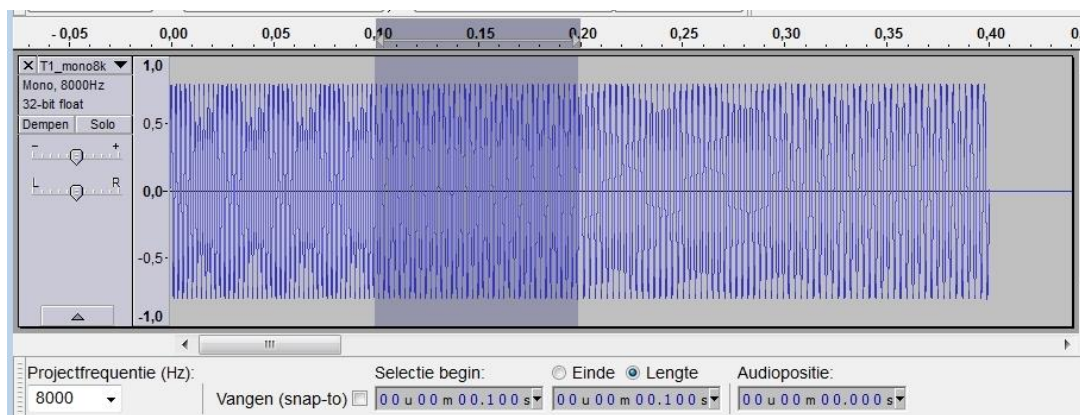
Ik gebruikte de volgend 25 fragmenten100 msec:

1	C	528 Hz
2	D	594 Hz
3	E	660 Hz
4	F	704 Hz
5	pause	silence
6	G	792 Hz
7	A	880 Hz
8	B	990 Hz
9	C	1056 Hz
10	pause	silence

11	C	1056 Hz
12	B	990 Hz
13	A	880 Hz
14	G	792 Hz
15	pause	silence
16	F	704 Hz
17	E	660 Hz
18	D	594 Hz
19	C	528 Hz
20	pause	silence

21	F	704 Hz
22	pause	silence
23	F	704 Hz
24	pause	silence
25	F	704 Hz

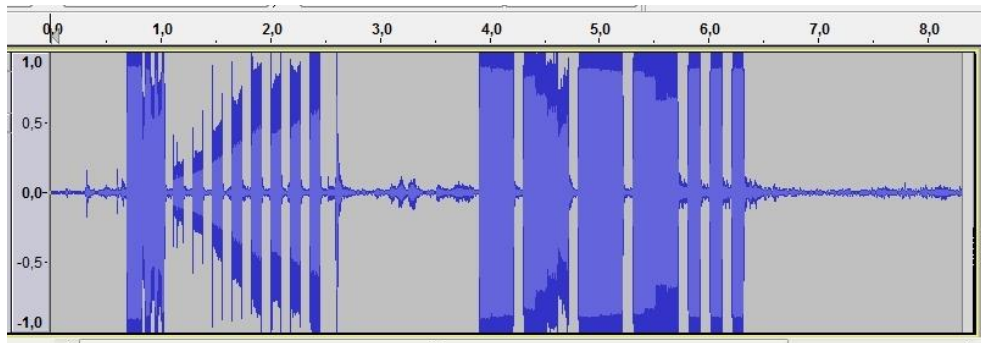
De stiltes geven een duidelijk beeld in de opname waar de prompt zit, en door de toonladdertjes is min of meer (het gaat erg snel) te horen wat er gebeurt.



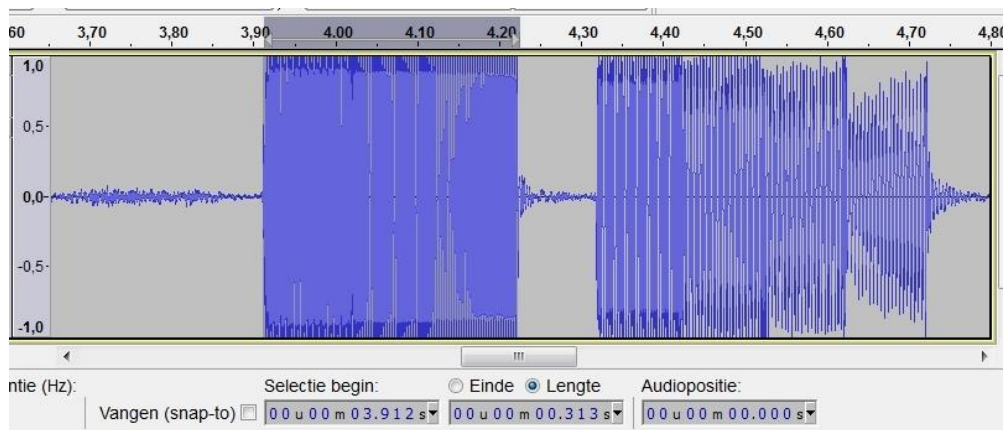
Figuur 7

Na aanbrengen van de WAV file was bij opbellen en afspelen (en opnemen met Audacity) duidelijk wat er fout ging, zie figuur 8 en 9.

Figuur 8 laat het opbellen zien. Van 1 tot 2,5 sec zijn de DTMF toontjes zichtbaar waarmee het testnummer werd 'gedraaid', vanaf 3,9 sec wordt het test-fragment afgespeeld. →



Figuur 8



Figuur 9

Figuur 9 zoomt hier op in, waaruit blijkt dat van de eerste 400 msec slechts 313 msec wordt afgespeeld. Blijkbaar verdwijnt de eerste 87 msec. Met dit gegeven kon de leverancier op zoek gaan naar de oorzaak, die snel werd gevonden en opgelost.

Al met al met ben ik zeer tevreden dat een opensourceprogramma, een standaard laptop en enige creativiteit tot zulke mooie en nauwkeurige testresultaten kunnen leiden. ←

TESTERS KUNNEN NIET INNOVATIEF ZIJN!

Door Joep Lobee • j.lobee@eclipseit.nl [@joeplobee](https://twitter.com/joeplobee) en Ard Kramer • a.kramer@eclipseit.nl [@ard_kramer](https://twitter.com/ard_kramer)



Het is volstrekt duidelijk waarom testers niet innovatief zijn. Hun werk bestaat uit het volgen van anderen: wachten op ontwerpen en software om deze te kunnen testen. In afwachting daarvan verzinnen ze wat testen en als ze deze op de software loslaten, dan melden ze terug of het aan de verwachting voldoet. Deze houding kan alleen betekenen dat testers nooit innovatief kunnen zijn!? Maar wat is innovatie eigenlijk? Een klein verhaaltje.

Wij, de schrijvers, waren met onze vriend Jan op vakantie in de Rocky Mountains, waar we op een grizzly beer stuitten. Wij keken elkaar aan en pakten snel onze hardloopschoenen uit onze tas. Jan keek ons aan en stelde de terechte vraag of we dachten dat ze de beer er uit konden lopen. Joep gaf hierop het simpele antwoord dat we zeker de beer er niet uit gingen lopen, maar dat we zo wel sneller waren dan Jan. En dat was dus het einde van Jan ...



Het gaat er dus om dat je vóórblijft op je concurrenten door middel van innovatie, anders verlies je de strijd. Vóórblijven kan soms slechts een paar stappen zijn en dan spreken we incrementele innovatie. Dit zijn innovaties die meestal verbeteringen of aanvullingen inhouden, de cruisecontrol in een auto bijvoorbeeld. Je kan ook een grote sprong voorwaarts maken en dan is er sprake van een radicale innovatie. Hierbij moet je denken aan de uitvindingen van de trein, de auto, en de computer.

Het waarom van innovatie is duidelijk, maar waaruit bestaat innovatie?

1. Een innovatie heeft betrekking op 'iets'. Hoewel we bij innovatie altijd het eerst denken aan producten, kan het ook gaan over nieuwe processen of diensten.
2. Verandering. Een innovatie betekent dat er een verandering komt. De mate van verandering kan verschillen zoals de voorbeelden van incrementele en radicale innovatie al aangaven. De eerste soort verandering is een bedoelde innovatie. Er zijn ook veranderingen die niet als innovatie bedoeld waren, maar het toch werden. Een voorbeeld van zo'n soort innovatie is de invoering van SMS. SMS was ontwikkeld voor snelle communicatie tussen technici. Dat het een enorm succesvol consumentenproduct zou zijn, voorzag niemand. Een derde vorm van verandering is serendipiteit: het vinden van iets onverwachts en bruikbaar terwijl je zoekt naar iets anders. De 'geeltjes' (post-its) zijn daar een voorbeeld van. De uitvinder was eigenlijk op zoek naar een superlijm.
3. Implementatie. Een innovatie is pas een innovatie als het wordt gebruikt. Een ondernemer zei eens: 'het is pas een innovatie als de tweede factuur is betaald'. Dit betekent dat je moet nadenken over hoe je de innovatie beschikbaar kunt stellen aan mensen en dat betekent: marketing.
4. Proces. Een innovatie bestaat uit een aantal noodzakelijke stappen, zoals het doorlopen van een businesscase en het zoeken van doelgroepen. De wijze waarop het proces loopt kan formeel zijn maar ook informeel, gestructureerd of juist ongestructureerd. Tegenwoordig ontstaat innovatie niet meer vanuit de zogeheten 'technology push' maar vanuit een 'market pull'. Innovatie is tegenwoordig veel meer gericht op hetgeen mensen willen dan op nieuwe technologie aan de man brengen. →

5. Brede scope van innovatie. Voor innovatie is vooral een brede kijk op zaken noodzakelijk. Zo kan het neerzetten van een merknaam innovatief zijn als je hiermee een compleet nieuwe afzetmarkt creëert. Een voorbeeld hiervan is Grolsch, dat zich in Engeland als zeer exclusief, met een bijzondere fles, wist te positioneren, een innovatieve aanpak.
6. Connectie tussen innovaties. Verbindingen maken tussen ideeën: dit worden ook wel systeeminnovaties genoemd. Een prachtig voorbeeld is de Senseo waar Philips en Douwe Egberts samen een nieuw soort koffieapparaat introduceerden.
7. Onzekerheid. Een belangrijk punt omdat je bij innovaties hier vooral mee te maken hebt en dan met name de factor tijd. Je kunt allerlei hobbels op de weg tegenkomen waardoor je niet zo snel je innovaties kunt introduceren als je zou wensen of dat innovaties niet zo aanslaan als je hoopte. Een voorbeeld hiervan is de introductie van video-telefonie in de jaren tachtig. Het idee was om elkaar te kunnen zien tijdens het telefoneren. Maar blijkbaar was de markt hier niet klaar en de benodigde techniek onvoldoende ontwikkeld. Dertig jaar later was de tijd wel rijp en werd het idee omgezet in succesvolle innovaties: Skype en webcams.
8. Creativiteit. Deze mag niet ontbreken, omdat we vaak creativiteit gelijkstellen aan innovatie, terwijl het slechts een onderdeel is van het complete palet. Creativiteit inzetten voor innovatie betekent ook een hoge kans op falen. Je creativiteit moet ook door anderen gezien en (h)erkend worden.



Als we alle punten overzien, kunnen we concluderen dat innovatie een creatief en onzeker proces is met een onzekere uitkomst. Daar nemen we als mensen natuurlijk geen genoegen mee en dan is de vraag gerechtvaardigd of innovaties moeten voortkomen uit creatieve, ongestructureerde of juist gestructureerde processen?

De ongestructureerde wijze

Een mooi voorbeeld van een innovatie die op ongestructureerde wijze tot stand kwam is penicilline. In 1928 zag Alexander Flemming, die onderzoek deed naar bacteriën, dat een van zijn petri-schalen was geïnfecteerd met een schimmel. Hij had de schaal kunnen weggooien: de bacteriecultuur was besmet en niet meer bruikbaar voor zijn onderzoek. Echter: hij keek nogmaals naar de schaal en zag dat er rond de schimmel geen bacteriën groeiden, de schimmel scheidde iets uit wat bacteriën doodde: penicilline.



Het zal overduidelijk zijn dat het product zeker een 'markt' zou hebben. Het duurde echter nog tien jaar voordat er serieus onderzoek naar penicilline werd gedaan en pas tijdens de Tweede Wereldoorlog in 1944 was penicilline een echt medicijn.

Dus: een goede ontdekking is een vereiste voor een echte innovatie. Maar wachten op een dergelijke ontdekking kan lang duren. Hoe vaak zou er eerder een met schimmel geïnfecteerde petrischaal zijn weggegooid? Dus als je als bedrijf dergelijke ontdekkingen kan stimuleren, is dat een potentiële bron voor innovatie. →

3M heeft het stimuleren van ontdekkingen expliciet gepromoot. Ze noemden dit 'Permitted Bootlegging'. Medewerkers mogen een bepaald percentage van hun tijd kunnen besteden aan werkgerelateerd onderzoek. Bij 3M gebruikte een chemicus deze tijd om te zoeken naar een superlijm. Hij ontdekte een lijm die eigenlijk compleet waardeloos was: het stolde niet en plakte ook niet heel sterk. Het enige was dat de lijm wat beter plakte als je hem aandrukte. Het was niet de superlijm die hij zocht, maar de onderzoeker was ervan overtuigd dat er een toepassing voor zijn lijm was, maar de grote vraag was: welke? Hij was lange tijd op zoek naar een toepassing voor zijn lijm. De oplossing had hij, nu zocht hij het probleem.

De toepassing kwam van een collega: die zocht naar een manier om hymnes in zijn kerkboek goed te markeren: verwijderbare plakkertjes zouden ideaal zijn. De twee kwamen bij elkaar en de post-it was geboren.

De term Permitted Bootlegging impliceert dat er ook non-permitted bootlegging is, oftewel: Bootlegging. Bootlegging wordt in de Business administration literatuur omschreven als: research waarin gemotiveerde employees in het geheim het innovatieproces organiseren, zonder officiële autorisatie van het verantwoordelijke management, in het belang van de organisatie.

De half gestructureerde wijze

Een bedrijf is er niet met tijd beschikbaar stellen en kijken wat er uitkomt. Lego heeft een andere aanpak gekozen. Lego is een enorm creatief bedrijf, het heeft innovatie en creativiteit een plaats gegeven in het 'lego vision lab'. Hier vinden creatieve koppen (liefkozend 'crazies' genoemd) een plaats gewijd aan creativiteit, er zijn zelfs speciale employees die als intermediair tussen de crazies en 'normale' mensen fungeren, wat ons erg aan de rol van testers doet denken.

Maar het vision lab is niet de enige reden waarom we Lego noemen: in 1998 lanceerde Lego Mindstorms. Een zelfbouw en zelfprogrammeerbare robot. Binnen twee weken was de firmware door hackers reverse-engineered, was er additionele software geschreven en waren er nieuwe sensoren gemaakt. Hiermee kreeg de robot veel meer mogelijkheden.

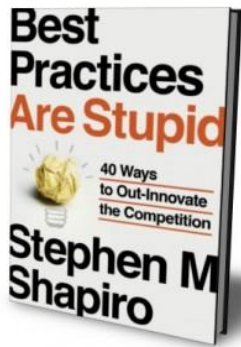
Lego had dit kunnen negeren of zelfs proberen tegen te gaan, maar ze deden iets radicaals: ze luisterde naar hun klanten. Ze omarmden de situatie en delen van de software en de home-brew sensoren werden in latere versies onderdeel van de robot-kit.

Bij Lego (en Google en 3M) is innovatie een onderdeel van de identiteit van het bedrijf, een deel van de dagelijkse business. Het is iets wat er gewoon bij hoort en dat is de sleutel om tot innovatie te komen, hoewel het proces nog redelijk ongestructureerd is. →



De gestructureerde wijze

Hoe ziet dan de gestructureerde wijze er uit? Dit wordt uitgelegd in het boek met de uitdagende titel 'Best Practices zijn dom'. Die titel zet je in ieder geval aan het denken. De moraal is dat als je best practices volgt, je nooit een innovator zult worden maar altijd een volger zult blijven. De auteur vertelt dat er een goede manier is om innovaties gestructureerd aan te pakken. De structuur, waar ook wij goede ervaring mee hebben opgedaan, gaat uit van het volgende simpele principe: een probleem breed bespreken en het terugbrengen tot één duidelijk gedefinieerd probleem, waarna je weer breed gaat nadenken over oplossingen die je weer terugbrengt tot de meest geschikte oplossing. En dat wordt dan de innovatie waar je voor gaat.



Het begint dus met het definiëren van het probleem dat je hebt. Einstein had dit al door. Hij zei niet voor niets dat als hij in 60 minuten een probleem moest oplossen, hij 59 minuten besteedde aan de analyse van het probleem, zodat hij de laatste minuut over de oplossing kon nadenken. Het gaat dus om het definiëren van het probleem dat je wilt oplossen en hiervoor moet je veel ideeën, meningen, ervaringen en dergelijke bij elkaar brengen. Heel vaak kiezen we hierbij voor brainstormtechnieken. Het bijzondere dat wij hebben geleerd over brainstormen is dat het juist een hele stille storm moet zijn. Mensen worden lui als anderen beginnen te roepen, of verliezen de focus omdat ze zich niet kunnen concentreren. Juist door een stille brainstorm ben je in staat om vele ideeën op te doen. Die ervaring hebben we opgedaan in een associatieoefening waar je aan je medecursist moest vertellen over een plaatje dat hij niet kon zien en je er alleen over mocht associëren. Hij moest zwijgen dus ik begon te praten. Na anderhalve minuut dacht ik dat ik wel alles had gezegd en viel de monoloog stil. Mijn medecursist mocht niets zeggen en er waren nog twee minuten over, dus zat er niets anders op dan door te praten en toen gebeurde er iets fenomenaals: ik boorde een nieuwe bron van associaties aan, die onuitputtelijk leek. Dus vooral stiltes inbouwen in de brainstorm is het motto. In het bepalen van een probleem kun je van tal van technieken gebruikmaken om op verschillende manieren problemen te definiëren zoals bijvoorbeeld door vijf maal de waarom vraag te stellen op een binnengekomen klacht.

Als je alle problemen op tafel hebt, moet je het tot één probleem terugbrengen en deze duidelijk definiëren en het terugbrengen tot een beschrijving in één zin. En dan ben je klaar en heb je een probleem. Gefeliciteerd! De oplossing van het probleem, waar halen we die? We beginnen weer met een brede benadering: zoveel mogelijk oplossingen bedenken. Hierbij kun je externe expertise inzetten. Zo zijn er in Nederland 'creatieve koppen' die op 'no cure, no pay' basis expertise inhuren voor een workshop om over oplossingen na te denken. Hierbij zetten ze juist niet-relevante expertise in omdat relevante expertise leidt tot blikvernaauwing: expertise als vijand van innovatie.

Vergroot ook je expertise door het inzetten van social media en internet. Op deze manier lukte het Proctor & Gamble om veel oude patenten nieuw leven in te blazen. Mensen op internet waren uitermate geïnteresseerd en behulpzaam om nieuwe toepassingen voor de patenten te bedenken. Het resultaat: Proctor & Gamble kon een deel van zijn R&D-afdeling sluiten en kreeg er veel externe expertise bij. Kijk overigens ook uit voor hoogopgeleide mensen: zij denken dat ze weten hoe het werkt en zullen daarom kansen afkraken. Als je dan ook nog oppast →

voor te veilige oplossingen (je moet soms grote stappen durven maken) en zorgt voor schaalgrootte die je past, dan ben je klaar om een keuze te maken voor een oplossing die je ligt.

Deze stappen hebben we een keer doorlopen met als conclusie: het was enkele dagen hard werken, maar het leidde wel tot een oplossing. En het was het leuk om te zien hoe onze oplossing zichtbaar werd in reclamespotjes van een grote bank.

Conclusie

Wat betekent dit alles nu voor ons testers? Hoe kunnen wij als testers onze innovatiekracht en creativiteit inzetten?

Om te beginnen kunnen we al bijdragen door innovaties te testen. Dit betekent wel dat je structuren moet durven los te laten en leren omgaan met onzekerheid en daar voor openstaan. Je kunt bijvoorbeeld je bijdrage leveren op internet als weekendtester. Houd altijd voor ogen houden wat de bedoeling is van het systeem dat je test: als je op deze manier naar de oplossing kijkt, dan heb je ook een goede focus op welke risico's of problemen je gedurende het testen kunt tegenkomen en hoe je hier mee moet omgaan.

Je moet je ook bewust zijn van hetgeen er dagelijks door je handen gaat als tester. Heb je er wel eens over nagedacht dat die ene bevinding die je hebt gevonden, misschien wel een oplossing is voor een (ander) probleem? Misschien niet de oplossing voor het systeem wat je aan het testen bent, maar kijk maar eens breder. Al eens overwogen om een lijst van nieuwe features in te leveren bij je opdrachtgever waarmee hij in de toekomst problemen mee kan oplossen? We moeten ons blikveld als testers niet vernauwen door bevindingen alleen op te vatten als 'fouten', omdat het afwijkingen zijn van de specificaties, maar ook als mogelijke features in toekomstige releases. Op deze manier moet je ook je eigen testwerk bekijken en zien hoe je daar verbeteringen kunt doorvoeren. Testautomatisering is tenslotte ook maar bedacht door testers die te lui waren om tests handmatig uit te voeren. En let op: verbeteringen die voor jou wellicht marginaal of onopvallend zijn, kunnen voor anderen vernieuwend zijn. Dus praat met collega testers over wat je doet en waar je mee bezig bent en wie weet kom je tot conclusie dat je met iets bezig bent waar anderen ook plezier van kunnen hebben: een innovatie.

Meedoen met innovaties moet ook een overlevingsstrategie zijn. Je wilt tenslotte niet de collega zijn die door de beer wordt opgegeten omdat anderen net iets sneller dan jij waren.

Dit artikel is gebaseerd op de presentatie van beide auteurs op EuroStar 2012 in Amsterdam ◀

EVENEMENTEN EN THEMA-AVONDEN

TestNet - Call for papers voorjaarsevenement

Deadline: zondag 23 december 2012

Het thema voor het voorjaar is 'Test Automation & tooling: silver bullets of testing?'

Testautomatisering werd in de recente TestNet programma enquête 2013 het meest genoemd als thema waarover de TestNet leden meer willen horen. In een tijd waarin agile een steeds belangrijkere plek inneemt, wordt test automatisering ook steeds belangrijker. Ook de opkomst van geavanceerde tooling, denk bijvoorbeeld aan model based testen, kan van grote invloed zijn op het werk van de tester. En wat is de invloed van open-source tooling op ons vakgebied? Allemaal vragen die een antwoord verdienen en mogelijk aan bod komen op het voorjaarsevenement van 2013. Deel jij je kennis en ervaring met je mede leden van TestNet? Kijk voor meer informatie op <https://www.testnet.org/details/84-callforpapers.html>

Thema-avond Crowdstesting

Dinsdag 15 januari 2013

Locatie: NBC, Blokhoeve 1, Nieuwegein

Thema-avond Kwaliteitsregie

Woensdag 6 februari 2013

Locatie: NBC, Blokhoeve 1, Nieuwegein

TestNet Noord Non-Functionals

Woensdag 13 februari 2013

Locatie: Hajé Heerenveen, Schans 65, 8441AC Heerenveen

Thema-avond Soft Skills

Dinsdag 19 maart 2013

Locatie: NBC, Blokhoeve 1, Nieuwegein

Thema-avond Testen van Mobile apps

Maandag 8 april 2013

Locatie: NBC, Blokhoeve 1, Nieuwegein

TestNet - Voorjaarsevenement 2013 TestTooling

Maandag 13 mei 2013

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