

TESTNET NIEUWS

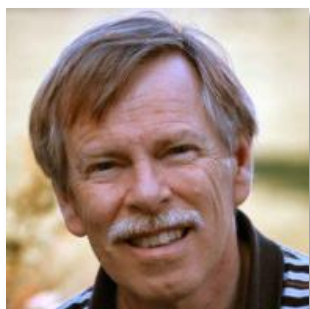
September 2012 • Jaargang 16 • Nummer 2

www.testnet.org secretaris@testnet.org



VAN DE REDACTIE

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



Normaal gesproken val ik jullie niet lastig met persoonlijke perikelen. Die zijn immers, zoals de naam al zegt, persoonlijk en dus niet interessant voor een breed publiek. Voor één keer wijk ik daarvan af.

Op het moment dat je dit leest, zit ik – hopelijk – op een tropisch strand onder een palmboom naar de zee te staren. Of, iets minder spannend, op een kantoor naar een pc-scherm.

Maar dan wel in Maleisië. Ja, beste lezer, de faam van het Nederlandse testen is inmiddels zo groot dat onze test-expertise tot een heus exportproduct is gesmeed. Dus als binnenkort de crisis is opgelost, dan weet je waardoor dat komt.

Wat nou zo leuk is (en daarom begin ik erover): tijdens mijn afwezigheid gaat TestNet Nieuws gewoon door. Dat bewijst eens te meer dat TNN een vitaal fenomeen is, waar een hoofdredacteur met gemak kan worden gemist.

Dank zij drie nieuwe redactieleden, die er met vol enthousiasme ingaan: Astrid, Guido en Rob. In deze TNN stellen ze zich aan jullie voor. Bestook ze van alle kanten met jullie eigen bijdragen, dan komt ook de volgende TNN weer snel vol! Vergeet ook de bestaande crew, Johan, Kees en Paul niet, want die zorgen voor de continuïteit. En via internet blijft jullie hoofdredacteur alles op een afstand volgen ...

Wil je mij volgen? Dat kan op <http://hans.vanloenhoud.eu/blog> ←

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>

IN DIT NUMMER

Verenigingsnieuws

Van de redactie	1
In dit nummer	2
Van de voorzitter	3
Riks Column: Agile is hot...	4
Vacature in het bestuur	5
Vacature in het bestuur	6
Twee vacatures in de evenementencommissie	7
Afscheid van het testen	8
Even voorstellen: Astrid Hodes	10
Even voorstellen: Guido Dulos	11
Even voorstellen: Rob van Steenberg	12
ISTQB nieuws	13
Interview met... Jeroen Vermeer	14
Mijn leukste / ergste testervaring	17
SCRUM en de onafhankelijke tester ...	18
HBO / Academische testopleiding	22
Agile Najaarsevenement wordt lenigste ooit	24
Open space op de TestNet Najaarsbijeenkomst	26

Artikelen testvakgebied

9,5 jaar Agile testen in Nederland	28
Is een testmanager nog nodig bij Agile?	30
Agile testen van datawarehouse & business intelligence	33
Uit het oog, uit het hart?	39
Testen op 81 vierkante meter	42
Agile testing isn't risking IT!	47
De rollen van de Agile tester	53
Het sprookje van de goede aanpak en de (sl)echte wereld	59
Lear lean testen, speel de Kanban game	63
As Agile as we can be	65
About testers and waste	66
Help! Mijn organisatie wil Agile...	72
Ervaren Agile testers wisselen best practices en randvoor-	75
Ergste misvattingen over testen bij Agile	80
Kun je Agile testen leren in een cursus?	83
Evenementen en thema-avonden	88

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

VAN DE VOORZITTER

Door Michiel Vroon • voorzitter@TestNet.org



Vakantietijd, heerlijk. Terwijl ik dit tik, zit ik er middenin. Deze periode van het jaar is wat mij betreft één van de fijnste om in te werken. Iedereen en alles is rustig, relaxed. Geen drukte op de wegen, altijd een parkeerplaats bij kantoor en zelden een snelheidscontrole, ook zij hebben recht op vakantie! Balans is overal. Mensen gaan richting hun vakantie weken of komen er net van terug en dat brengt vrolijkheid en luchtigheid met zich mee. Wat mij betreft mag iedereen onderweg blijven van of naar het vakantieadres. En dan hebben we ook nog de Olympische Spelen. Voor mij de slagroom op de zomerse taart. Geheel ontspannen kijk ik naar de fantastische prestaties die worden geleverd en waar vele jaren voorbereiding aan voorafgingen. Ik smul van de verhalen over toewijding, focus, drang, perfectionisme en eenzaamheid.

Mooi. Ik ga hier trouwens geen bruggetje maken naar ons mooie vak, dat laat ik aan jullie zelf over ...

Wanneer jullie dit lezen, staat het najaar voor de deur en beginnen de spelen weer voor ons. Met de verschillende disciplines die ons vak rijk is, probeert iedereen zijn stinkende best te doen. Er zullen winnaars zijn, er zullen verliezers zijn. Maar ook hier geldt: onderdeel uitmaken van de vereniging, meedoen is belangrijker dan winnen!

Ik wens iedereen veel plezier, inspiratie en wijsheid bij het lezen van deze TNN en op de komende thema-avonden, werkgroepbijeenkomsten en evenementen. ←

RIKS COLUMN: AGILE IS HOT...

Door Rik Marselis • rik@marselis.eu •  @rikmarselis



Agile is cool, Agile is nieuw, Agile is beter, Agile moet! Ben jij ook al lang Agile? Of denk je er niets van te weten? Of je het leuk vindt of niet, je kunt tegenwoordig niet om Agile heen. En als je een zichzelf respecterende IT-professional bent, dan moet je iets van Agile af weten. Al is het maar om de onzin die verteld wordt, eruit te kunnen filteren.

Veel testers vertellen mij dat ze nog niets van Agile af weten. Dan ga ik vragen stellen over hoe ze werken en wat ze in het verleden al hebben gedaan. Velen van hen komen dan, wanneer ze zich er in verdiepen, tot de conclusie dat ze best al wat van Agile weten.

Of dat ze, zonder het zelf te beseffen, al Agile gewerkt hebben. Zelf kwam ik er achter, dat ik 25 jaar geleden in een Agile team zat, maar de term bestond toen nog niet. Natuurlijk moet je wat nieuwe dingen weten, nieuwe termen (her)kennen en je de 'mindset' eigen maken. Maar daarnaast blijft veel bestaande kennis gewoon van toepassing of is de nieuw te verwerven kennis in grote mate 'gezond verstand'. Laten we eens kijken naar een paar misverstanden.

Misverstanden

De eerste is dat Agile een methode is. Niet echt, Agile is een 'mindset', 'a way of thinking'. Er zijn methoden die de Agile mindset (in meerdere of mindere mate) volgen.

Sommigen roepen, dat Agile volledig anders is. Dat valt volgens mij wel mee, de Agile mindset bevat heel logische en normale opvattingen. Het is bijvoorbeeld toch heel natuurlijk dat samenwerking belangrijker is dan contractonderhandelingen als je tot een gezamenlijk succes wilt komen! Maar samenwerking is ook in andere benaderingen van IT uiterst belangrijk. Ik zeg altijd maar 'Geef mij een klein team van heel goede mensen die elkaar aanvoelen en het maakt niet uit welke aanpak we volgen, er komt altijd wel een succes uit'. In de praktijk bestaan teams echter uit een mix van ervaren en onervaren mensen. Daarnaast varieert ook de intelligentie en motivatie. Kortom, een team met alleen maar toppers is niet reëel en ook niet nodig. Agile benadrukt samenwerking, daarmee kun je het beste uit de teamleden bundelen. En daarbij dwingt de Agile mindset je misschien wat meer om bewust met je vak bezig te zijn dan de aloude (trouwens ook vaak nog bijzonder goed toepasbare!) waterval aanpak.

Wat ik jammer vind, is dat Agile nogal eens als excuus gebruikt wordt om maar wat aan te rommelen. Het eerste, soms expres geïntroduceerde, misverstand is dat je niet meer hoeft te documenteren. Zinnige documentatie moet je altijd maken, maar niet overdrijven, een foto van een white board kan prima voldoende zijn. Een ander misverstand is dat bestaande testmethoden overboord kunnen, terwijl daar nog heel veel bruikbare kennis in zit. Ook in een Agile test is het van belang om risico's te classificeren en op basis daarvan je testzwaarte aan te passen en dus te variëren met testontwerptechnieken en -aanpakken.

Ervaring opdoen

Agile wordt nog niet lang op brede schaal toegepast, dus we moeten nog veel ervaring opdoen. Sommige mensen zijn huiverig, anderen zien het helemaal niet zitten. Maar zelfs als je het niets vindt, verdiep je er dan toch eens een beetje in. Kortom, mijn oproep aan elke tester die denkt niets van Agile te weten: Kom naar het TestNet evenement op 2 oktober en laat je verrassen hoe herkenbaar en toepasbaar de Agile mindset ook voor jou is!

(Lees je dit pas na 2 oktober, duik dan in de TestNet bibliotheek waar je de presentaties vindt.) ←

VACATURE IN HET BESTUUR

Door Bart Waterlor • secretaris@TestNet.org



Met ingang van de algemene ledenvergadering (ALV) van 2013 zal ik het bestuur van TestNet verlaten. Ik heb met veel plezier de functie van Secretaris vervuld, maar na zeven jaar vind ik het tijd voor andere hobby's. Dat betekent dat er met ingang van de ALV 2013 een vacature in het bestuur is voor de functie van Secretaris. Aangezien we belang hechten aan een naadloze overdracht wachten we niet tot de ALV met de werving, maar streven we ernaar om eind november een geschikte kandidaat te hebben gevonden die vanaf dat moment alvast meeloopt. Uiteraard verloopt de officiële benoeming via de ALV.

Wat houdt de functie van Secretaris in?

Als secretaris ben je verantwoordelijk voor de administratieve processen. Hieronder vallen taken als:

- bewaken van een correcte uitvoering van de ledenadministratie (uitbesteed aan A solution);
- beheren van het archief;
- notuleren bestuursvergaderingen en ALV;
- opmaken jaarverslag en jaarplan;
- opstellen jaarlijkse verenigingsagenda;
- communicatie met leden over lidmaatschap;
- communicatie met officiële instanties;
- bewaken van correcte informatieverstrekking naar leden over de vereniging.

Wie zoeken we?

We zijn op zoek naar een enthousiaste teamspeler met een dienstverlenende instelling, met organisatievermogen, met kennis van bestuurlijke processen en met goede mondelinge en schriftelijke uitdrukkingsvaardigheden.

Wat verwachten we van je?

Qua tijdsbesteding moet je denken aan gemiddeld twee uur per week, met pieken tijdens de eerste maanden van het jaar in verband met de voorbereiding van de ALV en de communicatie met leden rond de facturatieperiode. De klik met de overige bestuursleden is belangrijk voor de goede samenwerking.

Wil je graag een bijdrage leveren aan het succes van onze vereniging en denk je dat jij dé geschikte kandidaat bent voor de functie van Secretaris? Meld je dan aan voor deze functie door een mail te sturen naar secretaris@TestNet.org, voorzien van een CV en een korte motivatie. ←

VACATURE IN HET BESTUUR

Door Huib Schoots • wergroepen@TestNet.org



Met ingang van de algemene ledenvergadering (ALV) van 2013 zal ik het bestuur van TestNet verlaten. Dat betekent dat er een vacature in het bestuur komt voor de portefeuille 'werkgroepen'. Aangezien we belang hechten aan een naadloze overdracht, wachten we niet tot de ALV met de werving, maar streven we ernaar om eind november een geschikte kandidaat te hebben gevonden die vanaf dat moment alvast meeloopt. Uiteraard verloopt de officiële benoeming via de ALV.

Wat houdt de portefeuille werkgroepen in?

Je bent verantwoordelijk voor de gang van zaken rondom de werkgroepen. Momenteel is het bestuurslid werkgroepen ook lid van de evenementencommissie. Om continuïteit te garanderen heeft TestNet gekozen om twee bestuursleden plaats te laten nemen in de evenementencommissie. Over dit punt is eventueel overleg mogelijk binnen het bestuur.

De taken voor de portefeuillehouder werkgroepen:

- afstemming tussen werkgroepen en het bestuur;
- coördinatie oprichting nieuwe werkgroepen;
- rapportage van de voortgang, status en activiteiten van de werkgroepen binnen het bestuur;
- communicatie tussen de vereniging, de werkgroepen en het bestuur aangaande alle zaken rondom de werkgroepen;
- leiden jaarlijkse bijeenkomst(en) van de trekkers van de werkgroepen;
- coördinatie en invulling jaarlijkse thema-avond van de werkgroepen in november;
- (in overleg) lid van de evenementencommissie en de werkzaamheden in het kader van deze commissie; zie hiervoor de werving voor leden van de evenementencommissie.

Wie zoeken we?

We zijn op zoek naar een enthousiaste teamspeler met een dienstverlenende instelling, organisatievermogen, kennis van bestuurlijke processen en met goede mondelinge en schriftelijke uitdrukkingsvaardigheden. Een klik met de overige bestuursleden is belangrijk voor de goede samenwerking.

Wat verwachten we van je?

Qua tijdsbesteding moet je denken aan gemiddeld drie uur per week, met pieken tijdens in het voorjaar en najaar in verband met de voorbereiding van de evenementen en de thema-avond van de werkgroepen. Jaarlijks zijn er acht bestuursvergaderingen en twee evenementencommissie vergaderingen.

Wil je graag een bijdrage leveren aan het succes van onze vereniging en denk je dat jij dé geschikte kandidaat bent voor de portefeuille werkgroepen? Meld je dan aan voor deze functie door een mail te sturen naar wergroepen@TestNet.org, voorzien van een CV en een korte motivatie. ←

TWEE VACATURES IN DE EVENEMENTENCOMMISSIE

Door Rik Marselis • evenementen@TestNet.org



De evenementencommissie organiseert alle bijeenkomsten voor leden van TestNet, in het NBC in Nieuwegein, bij Hajé in Heerenveen of op locatie bij een testorganisatie. Gemiddeld is er één thema-avond per maand, twee keer per jaar een groot evenement en in de zomer de summer school. Met TestNet Noord in Heerenveen meegerekend staan er voor 2013 maar liefst zestien bijeenkomsten op het plan. De evenementencommissie bestaat momenteel uit acht leden. Om ook in de toekomst de bijeenkomsten succesvol te kunnen blijven organiseren breiden we de commissie graag uit met twee nieuwe enthousiaste leden.

Wat houdt de functie van evenementencommissielid in?

Gemiddeld twee keer per jaar ben jij eindverantwoordelijk voor een thema-avond. Samen met een back-up uit de commissie zorg je dat er een interessant programma over een testonderwerp komt. Je regelt sprekers, schrijft stukjes voor de TestNet nieuwsflits, de website en eventueel de TestNet Nieuws. Verder zorg je samen met andere leden van de commissie voor de registratie van bezoekers van een bijeenkomst. Onderling verdelen we het regelwerk zoals communicatie met de locaties, afrekenen via de penningmeester en meer van het organiseren dat bij zo'n bijeenkomst komt kijken.

Wie zoeken we?

We zijn op zoek naar enthousiaste mensen die het leuk vinden op een testevenement te zijn, die via verschillende kanalen contacten weten te leggen met mogelijke sprekers en die zelfstandig een leuke avond voor honderd collega-testers kunnen en willen organiseren. De leden van de commissie steunen en helpen elkaar waar nodig, maar je moet ook zelf initiatief nemen en zaken voor elkaar boksen.

Wat verwachten we van je?

Je tijdsbesteding ziet er als volgt uit:

- twee keer per jaar een thema-avond organiseren;
- twee of drie commissievergaderingen per jaar;
- gemiddeld eens per week contact per e-mail (rondom 'jouw bijeenkomst' uiteraard meer);
- bij de grote evenementen ben je aanwezig;
- bij thema-avonden ben je regelmatig aanwezig.

Kortom, gemiddeld ben je iets meer dan één avond per maand voor TestNet op pad en daarnaast doe je wat e-mail- en belwerk.

Wil jij graag een bijdrage leveren aan het de organisatie van de bijeenkomsten die de ruggengraat vormen van onze vereniging en denk je dat jij dé geschikte kandidaat bent? Meld je dan aan door een mail te sturen naar evenementen@TestNet.org voorzien van je CV en een korte motivatie. ←

AFSCHEID VAN HET TESTEN

Door Hylke ten Cate • Hylke.ten.Cate@planet.nl



Na een twaalf jaar testen ga ik mijn loopbaan afronden in de politiek. Dat ga ik doen bij de fractie of het wetenschappelijk bureau van 50PLUS. Een paar processen waar ik de laatste tijd mee te maken had, waren de invoering van een variabele pensioenleeftijd en de berekening van de restzetels binnen een lijstencombinatie.

De invoering van een variabele pensioenleeftijd werd op buitengewoon slordige wijze ingevoerd. Eigenlijk hadden in alle wetten, regelingen, contracten en software de constante 65 vervangen moeten worden door een variabele.

Zeker bij dat laatste hoort een heleboel testwerk. In de Eerste Kamer was een hoorzitting over de invoering. Daar kwam ook een vertegenwoordiger van de Sociale Verzekeringsbank aan het woord. Hier een paar citaten:

'Mevrouw Vermeulen: Bedankt voor de uitnodiging. Het is goed om te zien dat ook de Eerste Kamer oog en aandacht heeft voor de uitvoering van wetgeving. Mijn naam is Nicolý Vermeulen. Ik ben voorzitter van de Raad van Bestuur van de Sociale Verzekeringsbank (SVB). In die hoedanigheid ben ik eindverantwoordelijk voor de invoering en uitvoering van het onderhavige wetsvoorstel.

De SVB is de uitvoerder van volksverzekeringen in Nederland. We zorgen ervoor dat u, uw kinderen en uw ouders altijd en correct kinderbijslag, AOW-pensioen en nabestaandenuitkeringen ontvangen. Dat doen we al meer dan honderd jaar in opdracht van de overheid. De SVB heeft ruim vijf miljoen klanten, waarvan ruim drie miljoen AOW'ers. Zij rekenen op ons. Ze kunnen ook op ons rekenen, want wie een uitkering van ons ontvangt, maar ook wie informatie van ons wenst, is bij ons aan het goede adres. Daar staat de SVB voor. Daar sta ik ook voor.

Vanwege de korte reactietijd hebben we niet meteen kunnen reageren op de vraag wat de consequenties zijn voor de uitvoering. Daarom hebben we op 1 juni een aanvullende brief aan de minister gestuurd over de haalbaarheid van de invoering van het wetsvoorstel op 1 januari 2013. Wij moeten schuiven in onze ICT-capaciteit om dit wetsvoorstel voorrang te verlenen. De ICT-capaciteit op zich vormt met het huidige voorstel geen basis voor een risico. We moeten wel een aantal zaken verschuiven, vooral werkzaamheden die leiden tot besparingsverliezen. We hadden een aantal zaken gepland om kosten te besparen. Dat moeten we nu uitstellen. Het is meegenomen in het financieel kader van het voorstel. Het is dus verwerkt in datgene wat de Kamer gezien heeft.'

Hierop kwamen wel een paar vragen, zoals van Senator Noten.

'Mijn volgende vraag stel ik aan mevrouw Vermeulen. Zij is heel positief over de invoeringsmogelijkheden. De SVB heeft een heel flexibel systeem. Op 28 november 2011 hebben wij hier ook een verhoging van de AOW-leeftijd afgesproken. Een week later werd het desbetreffende voorstel aangenomen. Die verhoging heeft de SVB toen binnen enkele maanden kunnen realiseren. Vindt mevrouw Vermeulen het noodzakelijk om het nu te bespreken, of kan dat ook in september omdat het systeem van de SVB daarvoor flexibel genoeg is?'

Mevrouw Vermeulen: 'Ik ben positief gestemd, maar dat is nadat we alle risico's hebben afgewogen. We denken dat het mogelijk is om dit op 1 januari in te voeren. Natuurlijk hebben we er ook over nagedacht of we moeten wachten op besluitvorming over het wetsvoorstel. Een issue daarbij is de communicatie met de burger. →

We kunnen pas gaan communiceren met de burger als de wetgeving duidelijk is. Dat kan ertoe leiden dat we veel meer aanvragen en telefoontjes krijgen en dat we worden overspoeld door vragen waarop we geen antwoord kunnen geven. In het verleden is dat ook wel vaak gebeurd.'

De Eerste Kamer ging akkoord met deze redenering.

Over de ANP-software om de verkiezingsuitslag te berekenen is het nodige te doen geweest. De berekening wordt in twee slagen gemaakt. Eerst een verdeling over de twee lijstencombinaties en zestien losse partijen. Daarna volgt de zetelverdeling binnen de lijstencombinaties. Er gelden daar verschillende algoritmes voor het toewijzen van restzetels. Het ANP had gemakshalve voor beide hetzelfde algoritme gekozen. Dat hadden ze bij de vorige verkiezingen ook al gedaan. De bevinding werd vervolgens genegeerd. Misschien toch nuttig om die software te testen.

Zo zie je maar weer, dat je ook in de politiek te maken hebt met het testen van computerprogramma's.

http://www.eerstekamer.nl/behandeling/20120709/on gecorrigeerd_verslag_van_een/document3/f=/vj11rs75cxs8.pdf

Noot van de redactie: Hylke heeft jaren bijgedragen als redacteur aan TestNetNieuws. Met dit artikel neemt hij afscheid van TestNet. We wensen Hylke veel succes in de politiek! Ongetwijfeld hebben velen Hylke gezien op de televisie, bijvoorbeeld in de achtergrond bij Nieuwsuur in de aanloop naar de verkiezingen... ←

EVEN VOORSTELLEN: ASTRID HODES

Door Astrid Hodes • astridhodes@gmail.com



Tja, en dan heb je jezelf aangemeld als redacteur bij TestNet Nieuws. Absoluut nog niet kunnen overzien wat dat allemaal met zich meebrengt en dan krijg je als eerste opdracht: Schrijf een stukje over je zelf waar jij je aan het lezerspubliek voorstelt. Dat was even slikken. Mijn naam is Astrid Hodes, ik ben 46 jaar en al weer 25 jaar werkzaam in de ICT. Ik heb vele rollen en functies vervuld. Begonnen als Cobolprogrammeur, door naar systeembeheer vervolgens als troubleshooter en toen richting Project management.

Tja, en altijd was er daar het testen. Als een rode draad liep die door mijn werkzaamheden heen en op een dag besloot ik me daar meer in te verdiepen en voordat ik het wist was ik testcoördinator en daarna testmanager. Het leuke is dat ik nu profijt heb van alle rollen en functies die ik ooit heb vervuld. Ik begrijp de mensen en ik begrijp de processen. Momenteel ben ik werkzaam bij Bartosz ICT in de rol van senior testmanager en gelukkig zorgen zij nog steeds voor uitdagende opdrachten waardoor ik nog steeds met veel plezier mijn vak uitoefen.

In mijn vrije tijd lees ik, ga graag naar theater, uit eten of op vakantie. Ik ben getrouwd en heb twee jongens die af en toe heerlijk puberen waardoor je er weer aan herinnert wordt dat je ooit ook zo bent geweest.

Als redacteur van TestNet Nieuws hoop ik ook wat vernieuwing te kunnen brengen en lezers en leden zo ver te krijgen ook hun ideeën en ervaringen in een artikel te zetten. Wie weet bel of mail ik je binnenkort om jouw bijdrage te vragen. ←

EVEN VOORSTELLEN: GUIDO DULOS

Door Guido Dulos • GuidoDulos@valori.nl



Even voorstellen: Guido Dulos, 55 jaar. Hobby's: literatuur, schrijven, muziek luisteren, kletsen, wandelen, coaching en training. Geniet van het samen zijn met mijn vrouw en onze drie volwassen kinderen.

Op 1 september 1987 begon ik officieel mijn loopbaan in de ICT. Programmeren met de Spectrum 48K vond ik helemaal het einde en daarom maakte ik na mijn opleiding pedagogische academie en psychologie van die hobby mijn werk.

Na zeven jaar programmeur/analist/ontwerper/requirements engineer te zijn geweest bij Philips, maakte ik in 1994 de stap naar een klein softwarebedrijf. Daar was ik precies één dag tester en vanaf dag twee testmanager, nadat ik mijn nieuwe baas had uitgelegd dat testen iets meer om het lijf had dan achter het toetsenbord te gaan zitten en uit het hoofd controles uit te voeren.

In 2000 maakte ik de overstap naar Valori. Daar was ik tot 2004 testmanager. Vanaf 2004 ben ik trekker van Valori Academy en bedenk, maak en verzorg met mijn collega's trainingen, games en workshops, onder andere op het gebied van testen.

Als redacteur van TestNet Nieuws wil ik jullie helpen om op het prachtige platform TestNet Nieuws gevarieerde, leerzame, leuke en professionele artikelen te publiceren. ←

EVEN VOORSTELLEN: ROB VAN STEENBERGEN

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



Mag ik mij even voorstellen: Rob van Steenbergem, wonend te Lelystad tussen bos en water. Hobby's: Muziek, films, lezen, wandelen en eh... testen. Ooit ben ik begonnen als tester bij QualityHouse, toen we nog 'net niet allemaal' internet hadden, mobiele telefoons net geïntroduceerd waren voor het grote publiek. Ook de tijd dat we MS-DOS aan het verlaten waren. Interessant om te zien hoe de techniek met flinke snelheid is veranderd naar iets wat we nog niet konden bedenken. In die tijd was een les gevuld om alle facetten van een datumveld te leren testen en de regels te leren van schrikkeljaren.

Sinds die tijd heb ik het risico van problemen door viercijferige jaartallen, de migratie naar de euro, elektronische postcodeboeken, bankapplicaties, adresboeksystemen, afstandbedieningen, migraties van 8 bit naar 16 bit naar 32 bit applicaties, de OV-chipkaart en de virtualisatie van werkplekken getest. Testautomatisering met record & playback tools, testtechnieken gebruikt, TMap en andere methoden bestudeerd en gebruikt, zoals Agile en exploratory testen. Ik heb meegeholpen om testen te introduceren en te verbeteren bij bedrijven en testbegeleiding, testcoördinatie en testmanagement gedaan.

Door in de detachering en later als ZZP'er te werken heb ik dit bij veel verschillende soorten bedrijven gedaan, van bank- en verzekering, naar softwarehouse, naar technologische bedrijven als Philips en infrastructuur leveranciers. En in mijn vrije tijd doe ik ook steeds meer aan de 'test hobby'. Ik onderhoud een eigen website voor alle testevenementen over de hele wereld (www.testevents.com), schrijf een eigen blog en artikelen voor de Computable, testnieuws.nl en het blad 'TestingCircus'.

En nu dan redacteur bij het TestNet Nieuws magazine. Zoals je kan lezen verveel ik mij niet en zal ik met veel motivatie en creativiteit mijn bijdrage doen aan dit blad vóór en dóór testprofessionals. Misschien kom ik jou wel even aan je mouw trekken om iets te schrijven voor ons blad. Tot dan! ←

ISTQB NIEUWS

Door Bart Waterlor • secretary@bntqb.nl

ISTQB (International Software Testing Qualifications Board) bestaat in 2012 tien jaar. In deze periode hebben testers wereldwijd, ook uit Nederland, meegewerkt om een internationale, op best practices gebaseerd teststandaard neer te zetten. Op basis van deze teststandaard kunnen professionals zich laten certificeren. Het certificatieprogramma bestaat uit een Foundation level, een Advanced level en een Expert level. Inmiddels zijn er wereldwijd meer dan 240.000 certificaten behaald. In Nederland en België zijn dat er bijna vijfduizend, als volgt over de levels en modules verdeeld:

Foundation Level	2394
Advanced Level Test Management	1566
Advanced Level Test Analyst	475
Advanced Level Technical Test Analyst	242



De syllabi van de Advanced level modules zijn recentelijk onderhanden genomen. In het najaar komen daarvan nieuwe versies uit. Het Expert level is nog volop in ontwikkeling. De modules 'Improving the Test Process' en 'Test Management' zijn inmiddels gepubliceerd; de modules 'Test Automation' en 'Security Testing' zijn nog in ontwikkeling. Een wereldwijde teststandaard kan niet zonder een eenduidige definitie van termen en begrippen. Die zijn opgenomen in de Glossary. In het najaar verschijnt ook hiervan een nieuwe versie.

BNTQB (Belgium and Netherlands Testing Qualifications Board) vertegenwoordigt ISTQB in België en Nederland. BNTQB heeft onder andere een Engels-Nederlandse vertaling gemaakt van de Glossary. Vanzelfsprekend wordt ook de nieuwe versie van de Glossary vertaald. Daarnaast werkt BNTQB aan een Engels-Franse vertaling.



Meer informatie over het certificatieprogramma kun je vinden op de website van BNTQB, www.bntqb.org. Daar kun je ook de syllabi van de beschikbare levels en modules downloaden, evenals de Glossary en de vertaling daarvan. Ben je geïnteresseerd in het certificatieprogramma en ben je op zoek naar een partij die een training aanbiedt? Ook dat staat op de website van BNTQB.

De teststandaard van ISTQB is continu in ontwikkeling. Als je daaraan een bijdrage wilt leveren, dan kan dat. BNTQB is vertegenwoordigd in verschillende ISTQB-werkgroepen, onder andere voor de ontwikkeling van de verschillende certificeringslevels. Hierbij zijn ook testers uit Nederland en België betrokken.

Geïnteresseerd? Neem contact op met BNTQB via secretary@bntqb.nl. ←

INTERVIEW MET... JEROEN VERMEER

Door Rob van Steenberghe • rob@chickenwings.nl



In het verleden hadden we altijd het onderdeel 'Vijf vragen aan'. Dit is nu vervangen door 'Interview met...'; zodat we echt vragen kunnen stellen die bij de persoon past. Nu een interview met Jeroen Vermeer (J.Vermeer@rn.rabobank.nl), een freelance tester (bedrijf Sokaris).

Waar ben jij op dit moment mee bezig?

Op dit moment ben ik betrokken bij het programma SEPA binnen de Rabobank. Zoals je misschien weet, omhelst SEPA (Single European Payments Area) de eenwording van de Europese Unie op het gebied van betalingsverkeer.

Oftewel, alle banken binnen de eurozone moeten aan dezelfde standaarden gaan voldoen qua betalingen, incasso's en cards. Dit levert een enorme hoeveelheid werk op. Bij de Rabobank zijn er enkele honderden mensen die een bijdrage leveren aan het SEPA programma.

Mijn bijdrage is in de rol van testanalist/testcoördinator. Begin 2010 is er een pakket aangeschaft wat de zogeheten payment engine van de nieuwe betaalstraat gaat worden en mijn focus ligt op de implementatie van dit nieuwe pakket. Dit gebeurt in een reeks opvolgende releases (we hebben release 3 bijna afgerond). Voor een aantal van de releases ben ik betrokken (geweest) bij het uitwerken en uitvoeren van de UAT testset, bij het uitwerken en uitvoeren van deelketen testen en in een ondersteunende rol betrokken bij end-2-end testen. Doordat ik vanaf het eerste moment betrokken ben geweest bij de implementatie van het pakket heb ik een schat aan kennis kunnen opbouwen. Dat is de reden dat ik nu ook veel betrokken bij analyse van bevindingen en fungeer ik vaak als vraagbaak voor collega's.



Waar vind je de uitdagingen in deze opdracht?

Wat mijn drive is in deze omgeving, is de complexiteit, de sfeer en het uiteindelijke doel. Het complete nationale betalingsverkeer zal uiteindelijk (vanaf 2015) uitgefaseerd gaan worden en vervangen worden door SEPA. Dat betekent dus dat de nieuwe betaalstraat straks miljoenen transacties per dag moet gaan verwerken. Dus er rust een grote verantwoordelijkheid op ons werk. De betaalstraat is ook een enorm complex geheel van applicaties,



bankmedewerkers en processen dus ik leer elke dag nog nieuwe dingen. Het lijkt zo simpel een betaling van rekening 1 naar rekening 2, maar er komt toch heel veel bij kijken. Door de omvang van het programma werken een grote verscheidenheid aan teams en mensen mee, maar toch weet iedereen elkaar goed te vinden en wordt er intensief samengewerkt. Dit wordt zeker ook vergemakkelijkt door de werkwijzen die Rabobank heeft omarmt en de geweldige nieuwe locatie aan de Croeselaan in Utrecht. Je weet wel, de Verrekijker. →

Je bent al een tijd aanwezig in het bankwezen als tester. Is er een eenduidige manier waarop je testprojecten aanvliegt? Dat wil zeggen, heb je 'een methode, techniek of eigen manier' als je begint met een nieuwe opdracht?

De manier waarop ik testtrajecten aanvlieg is gebaseerd op een mix van intuïtie, ervaring en kennis opgedaan in trainingen en cursussen. Uiteindelijk wel een 'eigen manier', omdat mijn ervaring is dat elke omgeving toch weer anders is en het voor mij het beste werkt om eerst de omgeving goed in kaart te krijgen, zowel qua mensen als systemen. Mijn aanpak zou je dus top-down kunnen noemen; ik probeer voor mezelf eerst duidelijk te krijgen wat het doel van het project/programma is binnen de organisatie, welke systemen/applicaties erbij betrokken zijn, hoe de keten eruit ziet, welke sleutelfiguren er vanuit testoogpunt bij het project betrokken zijn, enzovoort.

Hierbij helpt bijvoorbeeld datgene wat ik van Prince2 geleerd heb en ook T-Map en ISTQB natuurlijk. Deze methoden zijn zeker niet heilig en allesomvattend voor mij, maar ik haal er wel bruikbare elementen uit. Ik zie mezelf als medewerker van het project als geheel, niet alleen als onderdeel van een team van testers, dus ik wil graag weten wat er allemaal speelt in het project.

Wat ik ook erg belangrijk vind, is het in kaart brengen waar ik welke informatie kan vinden. Zeker in grote organisaties zijn er vaak meerdere plekken aan te wijzen waar nuttige informatie zich bevindt. Het is zeker niet mijn doel om alles in mijn hoofd te krijgen, maar ik zorg er wel altijd voor dat ik weet waar ik wat kan vinden. Het aanleggen van een eigen mappenstructuur helpt me hierbij ook.

Deze aanpak is niet specifiek gericht op opdrachten in het bankwezen, en heeft dus het voordeel dat ik het op al mijn opdrachten kan toepassen.

Wat in mijn geval wel erg helpt, is dat ik zinvolle en zinloze informatie erg makkelijk kan scheiden en ook snel zaken kan opslaan in mijn geheugen. Dit heeft als voordeel dat ik in korte tijd veel kennis op kan bouwen, waardoor ik vaak als vraagbaak ga functioneren binnen het testteam en de rest van het project. Dit heeft wel als nadeel dat ik mezelf door mijn opgebouwde kennis al snel 'onmisbaar' maak, waardoor het soms lastig is om niet aanwezig te zijn en ik soms niet genoeg aan mijn eigen werk toe kom.

De studie die je gedaan hebt, is iets compleet anders dan je huidige vakgebied. Medisch bioloog, wat betekent dit precies en hoe helpt je dit in je huidige functie?

Wat betreft mijn achtergrond; een medisch bioloog is onderzoeker van het (dys) functioneren van het menselijk lichaam. Dit is alsnog heel breed natuurlijk, maar zelf heb ik mij tijdens mijn studie toegelegd op onderzoek naar de werking van de menselijke hersenen (cognitieve neurologie). Dit is een super interessant vakgebied, vandaar dat ik er ook voor gekozen heb. Naast dat het interessante materie is ben ik van aard een onderzoeker; ik wil graag weten hoe iets werkt en ik ga ook net zo lang door tot ik het weet.

Dit is ook de reden dat ik uiteindelijk in de testwereld terecht ben gekomen. Doordat er eigenlijk maar heel weinig werk was te vinden in de cognitieve neurologie, ben ik op een gegeven moment na mijn afstuderen een keer naar een carrièrebeurs gegaan om te kijken welke mogelijkheden er nog meer waren. Toen ben ik terecht gekomen bij het bedrijf waar ik zo'n vijf jaar voor gewerkt heb; Test Value (later KZA). Deze beroepskeuze bleek een schot in de roos, doordat ik met mijn achtergrond als onderzoeker ook uitstekend in staat blijkt te zijn om ICT-systemen →

te onderzoeken. Natuurlijk had ik nog wel het één en ander te leren toen ik begon bij Test Value, maar dat is vrij snel gegaan door trainingen, cursussen en de eerste opdrachten in het veld.

Achteraf ben ik er ook helemaal niet droevig om dat ik niet in de medische biologie ben doorgedaan; ik heb nu sowieso een veel afwisselendere baan die financieel gezien ook een stuk aantrekkelijker is. Als onderzoeker zit je toch al gauw jarenlang achter elkaar hetzelfde groepje cellen te onderzoeken voor een mager loontje. Wat aan de andere kant niets dan respect verdient, want dat zijn vaak wel de onderzoeken die ons leven een stuk aangenamer maken, omdat er medicijnen en hulpmiddelen uit voortkomen. Maar zoals gezegd, ik ben prima op mijn plek waar ik nu ben.



Hoe zie jij de toekomst van het testen?

Ik denk dat onder andere domeinkennis veel gevraagd zal gaan worden. Is nu natuurlijk ook al wel het geval, maar dit zal denk ik nog meer toenemen. Dit hangt denk ik samen met het feit dat er meer en meer Agile gewerkt gaat worden, waarbij er steeds minder gedocumenteerd wordt.

Daarnaast denk ik dat goede communicatieve vaardigheden steeds meer gevraagd zullen worden. Je ziet dat testtrajecten meer en meer over ketens van systemen gaan. Dit heeft als gevolg dat in vergelijking met het testen van een enkel systeem er met veel meer partijen afgestemd moet worden om tot een goed resultaat te komen.

Ook denk ik dat geautomatiseerd testen stukje bij beetje meer terrein zal gaan winnen, zei het wel erg langzaam, want het blijft lastig om de geldschieters te overtuigen om de investering te maken. Het outsourcen naar India zal denk ik ook afnemen, die bewegingen zie ik nu al ontstaan. Men komt er toch langzaam achter dat het niet de voordelen biedt waarop gehoopt werd. Al met al denk ik dat de vraag naar getrainde, ervaren testers alleen maar zal toenemen de komende jaren. Het zal dus een enerverend en uitdagend beroep blijven, denk ik. ←

MIJN LEUKSTE / ERGSTE TESTERVARING

Marcel Mersie • marcel.mersie@bartosz.nl



Hierbij lanceren wij een nieuwe rubriek 'Mijn leukste / ergste testervaring' die in de plaats is gekomen van de eerdere rubriek 'De dag van ...'. Wij hopen dat u net als ons er veel plezier aan beleeft.

Deze week vragen we Marcel Mersie naar zijn leukste/ergste testervaring.

Wat is jouw leukste/ergste testervaring?

Je voert een ketentest uit om te controleren of de door jouw opgevoerde bestelling daadwerkelijk door het gehele systeem wordt doorgevoerd. Vervolgens krijg je de volgende dag echt een gigantisch groot flatscreen tv thuis geleverd, welke je naar jouw idee toch echt op de testomgeving hebt besteld en niet in productie...

Wanneer speelde dit zich af?

Dit alles speelde zich af rond 2001. De tijd dat de combinatie internet, telefoon en schriftelijk bestellen nog in de kinderschoenen stond.

Kun je ons wat meer vertellen over deze ervaring?

De ervaring die ik daarbij heb opgedaan, is dat het moeilijk is en blijft om een goede testomgeving in te richten. Vooral als je gaat koppelen met andere systemen. Het is al niet eenvoudig om de gegevens tussen de systemen consistent te houden, zodat je ook daadwerkelijk je testgevallen kunt laten doorlopen. Maar het wordt nog eens moeilijker als je gaat koppelen aan daadwerkelijke productiesystemen.

Zorg ervoor dat je de omgeving helder in kaart brengt, niet alleen de testomgeving maar betrek ook de productie omgeving daarbij!

Waarom is dit jouw leukste testervaring?

Je krijgt niet iedere dag zo maar een gigantische flatscreen van de klant thuis gestuurd 😊

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

Ik zou niet direct meer een flatscreen bestellen, maar het eerst maar eens proberen met een paar sokken of zo. Die zijn wat gemakkelijker om weer mee terug te nemen naar de klant.

Aan wie geef je deze rubriek door?

Aan Reinier van der Pas. ←

SCRUM EN DE ONAFHANKELIJKE TESTER...

Door Bryan Bakker • bryan.bakker@sioux.eu •  @Bryan_Bakker



Als ik een kijkje neem op Wikipedia bij de artikelen over Agile en Scrum, dan staat het woordje 'test' toch maar zelden vermeld. En dan eigenlijk ook alleen in de context van test-driven development.

Juist deze test activiteit ligt meestal niet bij testers. Zijn testers daarmee overbodig in een Agile omgeving? Moeten wij testers de opkomst van Agile zien als bedreiging?

Een kans

Laat ik meteen duidelijk zijn, Agile of bijvoorbeeld Scrum is geen bedreiging voor het testvak, maar juist een kans. Testers roepen al decennia lang dat ze eerder bij projecten betrokken willen worden, zodat al vroeg over het testen nagedacht kan worden en niet alles achteraf moet gebeuren. Wij willen zo snel mogelijk beginnen met testen, ook als de software en het systeem nog lang niet klaar zijn. Juist in Agile omgevingen zou dit eenvoudig mogelijk moeten zijn. Toch hebben veel testers moeite om een passende rol hierin te vinden.

Bij Sioux Embedded Systems wordt software ontwikkeld en getest voor klanten uit technische omgevingen. Daarbij hebben ontwikkelaars en testers niet alleen met software te maken, maar ook met andere disciplines zoals elektronica, firmware en mechanica. De integratie met deze andere disciplines vergroot de complexiteit van het gehele project aanzienlijk. Steeds meer klanten gebruiken een Agile manier van werken (meestal in de vorm van Scrum). En ook alle interne projecten bij Sioux worden met behulp van Scrum uitgevoerd. De verschuiving naar Scrum heeft ook impact op het testen...Test-Driven Development.

Binnen Scrum projecten is er veel aandacht voor testen. De ervaringen met test-driven development zijn zeer goed. Eerst worden de (automatische) testen gedefinieerd, en daarna wordt pas de code zelf geschreven. En als het goed is, slagen de testen dan. Anders heeft de ontwikkelaar nog meer werk. De set van testen wordt groter naarmate meer functionaliteit wordt geïmplementeerd en fungeert daarmee als krachtige regressie test (op component niveau).

Ontwikkelaars zullen regelmatig refactoring op bestaande code uitvoeren, bijvoorbeeld omdat nieuwe features anders niet efficiënt geïmplementeerd kunnen worden, of omdat er te veel technical debt is ontstaan. Na een refactoring slag zal de ontwikkelaar de regressietest gebruiken om aan te tonen dat bestaande functionaliteit nog steeds naar behoren werkt. Zonder een degelijke (automatische) regressietest bevatten refactoring activiteiten grote risico's; omgevallen functionaliteit wordt immers niet meteen gedetecteerd.

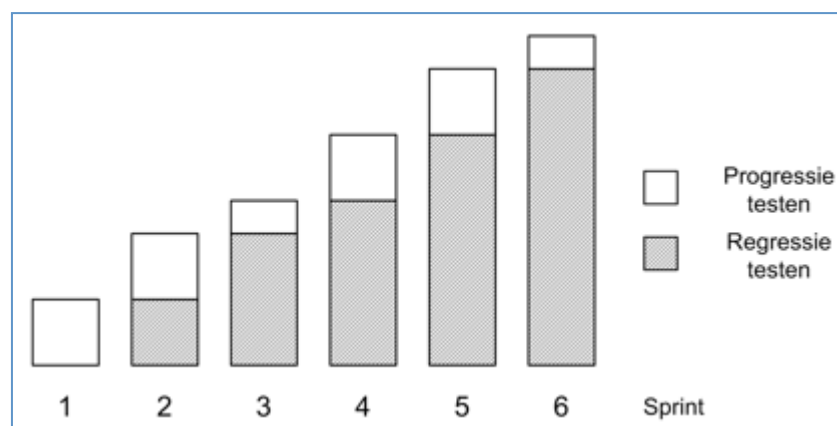
Scrum tester

De testen bij test-driven development worden gedefinieerd door de ontwikkelaar zelf. Hierbij bestaan echter enkele risico's. Ontwikkelaars kijken vooral naar de happy flow en zijn daarnaast blind voor hun eigen fouten (zoals iedereen). Tevens wordt de code zodanig geschreven dat de testen slagen, ook dat is niet zonder gevaren. Een tester kan hier juist veel toegevoegde waarde bieden, door op een andere manier naar de geleverde functionaliteit te kijken.

Testers hebben de neiging om naar andere paden in de software te kijken, zoals foutafhandeling, of alternatieve paden om een feature te gebruiken, vaak in combinatie met testtechnieken. Het biedt veel toegevoegde →

waarde een tester toe te voegen aan het Scrum-team en dit soort testen uit te laten voeren. Binnen het Scrum-team hebben dan zowel de tester als de ontwikkelaar een testverantwoordelijkheid.

De tester in het Scrum-team zal nauw met de betreffende ontwikkelaars communiceren en is direct op de hoogte van de status van leveringen binnen de huidige sprint. De tester kan de nieuwe functionaliteit van deze sprint meteen binnen de sprint verifiëren. Bij het werken in sprints wordt het aandeel regressie testen steeds groter, zie Figuur 1. Progressietesten zijn de testen die nieuwe functionaliteit verifiëren, regressie testen controleren dat al bestaande functionaliteit nog steeds goed functioneert. Het automatiseren van testen is dan ook zeer belangrijk. Omdat de tester in hetzelfde team zit als de ontwikkelaars, zal dit vaak voorspoediger verlopen. Niet alleen van de ontwikkeltesten maar ook van de overige testen. Ontwikkelaars kunnen meedenken en meehelpen met het automatiseren. Zo bestaat de (automatische) regressie test niet alleen uit de testen vanuit het test driven development principe, maar ook uit functionele testen die door de tester worden gedefinieerd.



Figuur 1

Door Scrum teams uit te breiden met een tester zullen de ontwikkelaars snel feedback krijgen over de nieuwe features, binnen dezelfde sprint worden de fouten al gevonden. Nadeel is wel dat testers minder onafhankelijk worden, en zo mee worden genomen in de waan van het ontwikkelproject. Dit hoeft op zich geen probleem te zijn, maar men moet zich er wel bewust van zijn. Veel testers ervaren het als zeer positief om dicht bij de ontwikkelaars in korte sprints te werken, en willen vaak niet meer anders. Ze voelen zich veel meer als echt onderdeel van een team en zien hun toegevoegde waarde snel terug in het product.

Onafhankelijke tester

In niet-Agile omgevingen werken testers vaak onafhankelijk van ontwikkeling. Het idee om testers in Scrum-teams te plaatsen is zeker geen slecht idee: De snelheid waarmee ontwikkelaars feedback van de testers krijgen is vele malen sneller dan in niet-Agile projecten.

Sommige testers hebben juist de kracht om met enige afstand naar ontwikkelingen in een project te kijken en adequate correctieve acties te ondernemen. Door direct in het Scrum-team te opereren, wordt dit een stuk moeilijker en het risico bestaat dat deze toegevoegde waarde verdwijnt.

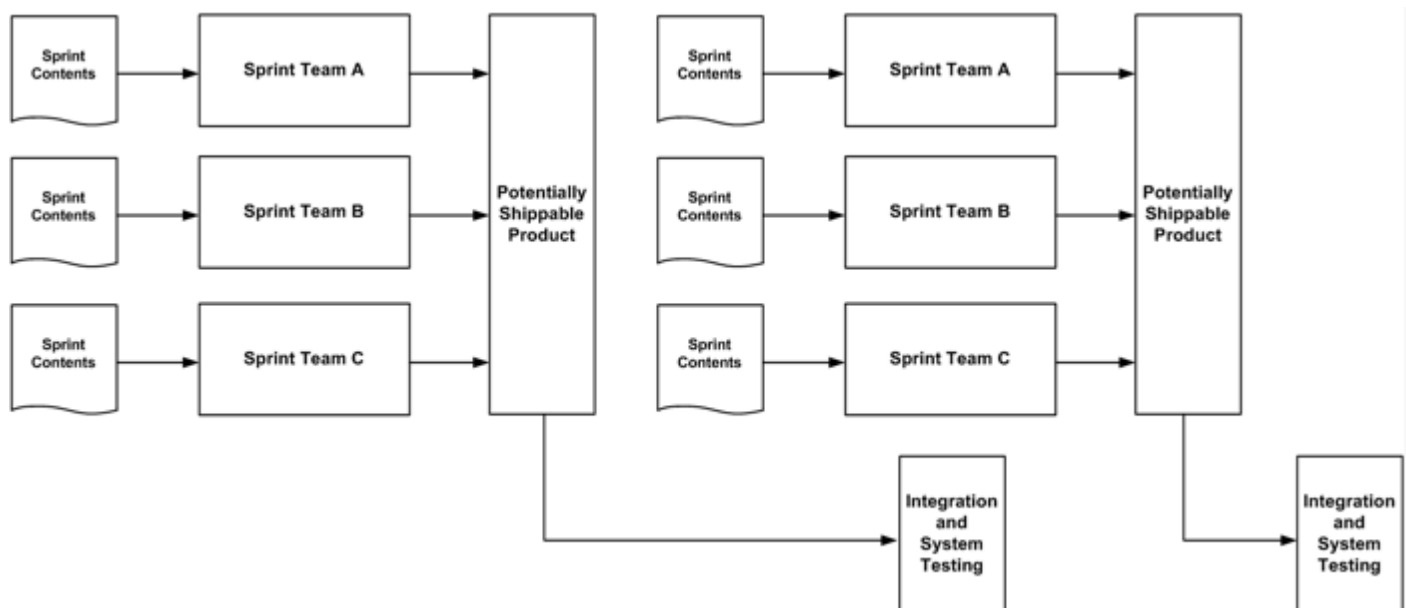
Vooral bij grotere projecten met meerdere disciplines kunnen dergelijke testers nog steeds grote toegevoegde waarde hebben, maar dan vooral buiten de Scrum-teams. We zien bij veel projecten nog steeds behoefte aan 'onafhankelijke' testers. Deze kijken juist naar integraties met andere disciplines en controleren functionaliteit op systeem niveau. Als Scrum-teams hun functionaliteit op orde hebben, kan deze functionaliteit op hoger niveau met een onafhankelijke view getest worden. Vaak beginnen deze activiteiten iets later in het project, dus zeker →

niet volgens de Scrum ideeën. Soms is het echter zeer moeilijk aan de Scrum manier vast te houden als hardware of mechanica nog niet beschikbaar zijn en men met simulaties moet volstaan. Daarnaast vinden integraties en systeemtesten vaak plaats op systemen (bijvoorbeeld MRI scanners of waferscanner) die uiterst schaars en duur in gebruik zijn en daarnaast niet altijd beschikbaar zijn.

Simulaties en emulaties

Als bepaalde systeem onderdelen of configuraties pas laat in het project beschikbaar zijn, dan moet het uiteindelijke systeemtesten van deze functionaliteit langer wachten. Simulaties en emulaties bieden wel de mogelijkheid eerder zaken gedeeltelijk te testen en eerder fouten te vinden, maar de uiteindelijke test heeft toch de volledige configuratie nodig.

In figuur 2 is een overzicht te zien van drie Scrum-teams die functionaliteit opleveren in een potentially shippable product (PSP). Dit PSP gaat na de sprint door naar het Integratie en Systeem Test Team om op echte hardware getest te worden. In dit geval lopen de integratie en systeemtest activiteiten dus één sprint achter. Vaak zien we ook dat niet elke PSP aan het einde van een sprint gebruikt wordt door het Integratie en Systeem Test Team, omdat dergelijke testen vaak lang duren en overstappen naar een nieuwe versie van de software veel tijd in beslag kan nemen. Het installeren van nieuwe software op complexe systemen (zoals waferscanners of elektronenmicroscopen) kan veel tijd kosten. Daarnaast is het vaak niet nodig altijd de nieuwste versie van de software op de systemen te hebben voor lopende test activiteiten. De feedback naar ontwikkeling is dan wel langzamer dan van de test activiteiten binnen het Scrum-team zelf.



Figuur 2

Scrum blijft ook in dergelijke omgevingen een goede manier van werken. Echter, zal er niet na elke sprint een potentially shippable product zijn, wel shippable naar een andere groep binnen de organisatie maar niet naar een eindklant. De definition of done zal hierop aangepast kunnen worden om toch functies in een sprint te kunnen afronden. Het Integratie en Test Team kan zelf uiteraard ook in sprints werken.

Ook bijvoorbeeld safety testen of EMC (Elektromagnetische compatibiliteit) metingen zullen niet met elke sprint opnieuw uitgevoerd worden. Daarvoor zijn ze veel te kostbaar en te langdurig. Deze worden op de finale →

versie van het systeem uitgevoerd en vaak ook eerder, maar zeker niet met elke sprint. In sommige omgevingen is (uitgebreide) documentatie zeer belangrijk, voor bijvoorbeeld bewijslast van test resultaten voor safety standaarden. De onafhankelijke testers in het Integratie en Test Team zijn uitermate geschikt voor dit soort test- en documentatie activiteiten. Zij hebben het grotere overzicht, veel meer dan de testers binnen de Scrum teams.

Conclusie

In Scrum projecten is het toevoegen van testers aan de Scrum-teams een goede manier om eerder feedback te krijgen op de ontwikkelde functionaliteit. Echter, zijn onafhankelijke testteams nog steeds nuttig en zelfs noodzakelijk in specifieke omgevingen, deze kunnen wel aanzienlijk kleiner zijn dan in traditionele (niet Scrum/Agile) projecten.

Van de redactie

Bryan Bakker zal tijdens het TestNet Najaarsevenement op 2 oktober 2012 een lezing geven met de titel: 'Scrum and testing... the end of the test role?' ←

HBO / ACADEMISCHE TESTOPLEIDING

Door Jos van Rooyen • jos.van.rooyen@bartosz.nl



Sinds de jaren '90 van de vorige eeuw wordt er intensief gewerkt aan onder andere de ISTQB certificering. Daarnaast worden er binnen diverse bedrijven specifieke trainingen of masterclasses ontwikkeld op het gebied van testen. Ook binnen diverse ICT-opleidingen komt het vak testen in verschillende vormen terug, bijvoorbeeld als minor. Een nadeel van deze aanpak is dat er steeds deelgebieden van het testvak worden onderwezen en vaak slechts op globaal niveau.

De afgelopen jaren zijn op deze manier honderden mensen opgeleid tot tester.

Daarnaast heeft het testvak zelf in de afgelopen jaren een belangrijke groei in volwassenheid doorgemaakt. En juist omdat testen een vak is mag een professionele testopleiding niet ontbreken. Een dergelijke integrale Testopleiding op HBO / Academisch niveau binnen Nederland maakt het mogelijk de volgende stap te zetten in de volwassenheid van het testvak.

Initiatief

Vandaar dat Huib Schoots en Jos van Rooyen het initiatief hebben genomen om een werkgroep te starten om de mogelijkheden te onderzoeken voor een testopleiding op HBO / Academisch niveau. Huib en Jos zijn al vele jaren zeer actief in het testvak in diverse rollen. Zij zijn ervan overtuigd dat een dergelijke opleiding een absolute must is in de verdere ontwikkeling en professionalisering van het testvak.

Dat wij niet het eerste land zijn die invulling willen geven aan deze behoefte blijkt wel uit het initiatief in de Scandinavische landen. Daar wordt al sinds enige jaren gewerkt aan een dergelijke opleiding.

Doel werkgroep

Het doel van de werkgroep is de mogelijkheden en haalbaarheid te onderzoeken van een HBO / Academische Testopleiding in Nederland. Het doel valt in een aantal delen uiteen, te weten:

- inventarisatie van de huidige gedoeerde stof;
- een inventarisatie naar de reële behoefte (zowel vraag- als aanbodzijde);
- een voorstel voor het curriculum van een dergelijke opleiding;
- een implementatieplan voor deze opleiding.

Inmiddels heeft de werkgroep met dertig enthousiastelingen de kick-off gehouden en besloten door te gaan. De leden van de werkgroep zijn afkomstig uit diverse organisaties zoals opleidingsinstituten, overheidsorganisaties en consultancybedrijven. Daarnaast hebben veel leden ervaring als docent, gastdocent of lector op onderdelen van het testvak. Tevens is een aantal leden betrokken of betrokken geweest bij het opzetten van opleidingen in de breedste zin van het woord. Kortom, een mixture waar alle ingrediënten aanwezig zijn om een mooi resultaat met elkaar te ontwikkelen.

Tijdens de eerste bijeenkomst kwam regelmatig de vraag naar voren waarom er relatief weinig aan testen wordt gedaan? Een vraag waar we nog geen eensluidend antwoord op konden geven. Als er TestNet leden zijn die daar een passend antwoord op hebben dan zijn wij als werkgroep nieuwsgierig naar het antwoord. →

Een tweede punt dat als rode draad door de bijeenkomst ging was het grote verschil aan behoefte. Is er daadwerkelijk behoefte aan een HBO / Academische testopleiding? Regelmatig kwam ook naar voren dat op MBO niveau ook behoefte is aan een testopleiding. Daarnaast kwam de vorm van een testopleiding regelmatig terug. Als specialisatie, of minor of een separate opleiding waren een aantal gedachten die langs kwamen. Kortom, genoeg vraagstukken om de tanden komende periode in te zetten en verder uit te werken.

Hoe ziet de komende periode er uit?

Als eerste gaan we kijken wat er allemaal gedoceerd wordt momenteel op de diverse opleidingsinstellingen. Alle leden hebben dit als huiswerk meegenomen om in hun eigen netwerk een inventarisatie uit te voeren. De tweede vraag waar we als werkgroep over nadenken is hoe het ideale curriculum er uit moet zien om een goede testopleiding op te zetten. Op basis van deze twee actiepunten gaan we in september verder.

Het idee is om de werkgroep gedurende één jaar in stand te houden en dan een concept aan te bieden aan Test-Net als onafhankelijk platform. Alle instellingen enzovoort kunnen daar dan gebruik van maken om hun eigen opleiding uit te breiden dan wel op te zetten.

Tot zover de update. Voor vragen kun je met ondergetekende contact opnemen. We zullen regelmatig een update verzorgen. ←

AGILE NAJAARSEVENEMENT WORDT LENIGSTE OOI

Het is al bijna zover: het Najaarsevenement met het thema 'Hoe worden we nu echt Agile testers?'

Op dinsdag 2 oktober 2012 komen we in het NBC in Nieuwegein bij elkaar voor het volste evenementenprogramma ooit bij TestNet. Nog nooit hadden we activiteiten in zo veel zalen tegelijk. Nog nooit was er zoveel keuze. En nog nooit kon je zelf zo actief met je vak aan de gang of over je vak in discussie. In het testlab kun je zelf testen. Leer omgaan met testtools in de workshops van Dennis Huizing en Martin Gijsen. Doe een workshop planning poker. De open space (op het balkon van de Grand Hall, meer open konden we de space niet vinden) geeft je alle ruimte om zelf een belangrijk onderwerp onder de aandacht te brengen en daarover in discussie te gaan (zie de beschrijving elders in deze TestNet Nieuws). Natuurlijk hebben we ook de bekende elementen in het programma: Vijf Tutorials in de ochtend, een keynote aan het begin van de middag en begin van de avond. En zestien parallel presentaties in de subzalen.

Een impressie van het programma:

Tutorials:

Dit keer staan maar liefst vijf tutorials op het ochtendprogramma: Egbert Bouman en Chris Maliepaard leren je op spelgebaseerde wijze wat Kanban is en hoe je het als tester moet toepassen; Anko Tijman en Marnix van den Ent geven je in 'een dag uit het leven van een Agile tester' een totaaloverzicht van de rollen en activiteiten die horen bij een tester in een Agile setting. In 'Mind Maps, an Agile way of working' leren Jean-Paul Varwijk en Huib Schoots je om als tester mee te gaan in de lagere documentatiebehoefte van Agile-projecten, zonder dat dit ten koste gaat van de inhoud. Cecile Davis en Leo van der Aalst vertellen je hoe je de teststrategie en documentatie aanpast aan een Agile traject. Als je meer wilt leren over 'Combinatorial testing' (diverse vormen van pairwise testing) zit je goed bij Peter Zimmerer.

Keynotes:

Wij zijn verheugd dat we een van de grondleggers van het Agile Manifesto bereid hebben gevonden om op het najaarsevenement een keynote-presentatie te verzorgen: Arie van Bennekum zal in zijn presentatie zijn visie op het Agile manifesto toelichten, en specifiek ingaan op de betekenis ervan voor testen.

De avond keynote wordt verzorgd door Peter Zimmerer. Hij was in 2003 testmanager van het allereerste Agile project van zijn werkgever Siemens AG in München. Sindsdien heeft hij een schat aan ervaring opgedaan in de vele Agile projecten die volgden en op basis daarvan kijkt hij met ons terug op elf jaar huwelijk tussen Agile en testen.

Presentaties:

Nog niet eerder waren er in Nederland op één dag zoveel presentaties over Agile en testen. Uiteenlopend van kijken naar historie en toekomst, parallel met coaching in de sport, testen bij de spoorwegen, tot luchtiger vertoningen als een optreden van Laurel & Hardy of het bemoeitheater! ➔

Stel je eigen programma samen op basis van de informatie op de website, daar vind je ook de actuele versie van het programma (tip: sla deze JPG op je smartphone op).

Tot ziens op het Najaarsevenement!

Versie 1.0	TestNet Najaarsevenement: "Hoe worden we nu echt Agile testers?"				dinsdag 2 oktober 2012
begintijd	Ochtendprogramma: TUTORIALS Uitsluitend toegankelijk voor TestNet leden				
	Zaal A	Zaal B	Zaal C	Zaal D	Zaal E
9:00	Ontvangst / Inschrijving / Koffie				
09:30 (rond 11:00 30 minuten koffie- pauze)	Egbert Bouman en Rob Verheul Test Kanban, speel de game! (max. 50 deelnemers)	Anko Tijman en Marnix van den Ent Een dag uit het leven van een Agile tester (max. 50 deelnemers)	Jean-Paul Varwijk en Huib Schoots Mind maps: an agile way of working (max. 50 deelnemers) NEEM LAPTOP OF TABLET MEE!!	Cecile Davis en Leo van der Aalst (fr)Agile Balance (max. 50 deelnemers)	Peter Zimmerer Combinatorial Testing Explained (max. 50 deelnemers)
13:00	Lunch voor tutorialbezoekers en standbemensing				

Versie 1.0	TestNet Najaarsevent: "Hoe worden we nu echt Agile testers?"					dinsdag 2 oktober 2012
begintijd	Middagprogramma					
13:00	Ontvangst / Inschrijving / Koffie / Informatiemarkt					
14:00	Opening door de voorzitter Agile !! Introductie, uitleg & instructie Open Space					
14:20	Arie van Bennekum Forward reasoning or learning from the past. Where the Agile Manifesto helps.					
15:10	Koffiepauze en informatiemarkt					
	Grote zaal	Zaal 8-10	Zaal 11 - 12	Zaal 14	Zaal 5/6/7 Agile Workshops	Balkon: Open space
15:40	Anko Tijman 9,5 jaar Agile testen in Nederland - en hoe het verder moet	Bram Bronneberg Agile testing isn't risking IT	Ard Kramer Testen op 81 m2	Iris Groenewoudt en Armando Dörsek Agile testen van Business Intelligence	tutorial: Dennis Huizing Starten met Selenium (neem je eigen computer mee !!)	Open Space
16:25	Zaalwissel					
16:35	Erik Boelen Mijn toolbox als een agile test manager	Jeroen Mengerink Agile en outsourcing: gaat dat wel samen?	Egbert Bouman Ervaringen met een goede agile aanpak	Bryan Bakker Scrum and testing ... the end of the test role?	Workshops: Zie het bord met aankondigingen	
17:20	Diner en Informatiemarkt					
	Avondprogramma					
18:30	Peter Zimmerer Lessons Learned in Agile Testing @Siemens					
19:20	Zaalwissel					
	Grote zaal	Zaal 8-10	Zaal 11 - 12	Zaal 14	Zaal 5/6/7 Agile Workshops	Balkon: Open space
19:30	Jurian van de Laar Leren agile testen, kwestie van cursus volgen?	Stefaan Luckermans en Dominic Maes About testers and garbage men (testen met Stan Laurel en Oliver Hardy)	Mirjam Swets, Rik Teuben, Martin Heynen Bemoeitheater: As Agile as we can be	Sander de Jonge De sprinter of toch de noodrem? Agile testen bij de NS	tutorial: Martin Gijzen FitNesse en Selenium voor testers en test automatiseerders (neem je eigen computer mee !!)	Open Space (bij voldoende belangstelling)
20:15	Zaalwissel					
20:25	Arjen Verweij Agile: testing effectively in a complex environment	Yvonne van Zaanen Uit het oog, uit het hart	Pascal Dufour Ik zie ik zie wat jij niet ziet	Paul Custers Help! Mijn organisatie wil Agile ...	Workshops: Zie het bord met aankondigingen	
21:10	Borrel en informatiemarkt					



OPEN SPACE OP DE TESTNET NAJAARSBIJEENKOMST

Door Johan Vink • vink.jd@gmail.com



In de Agile wereld is communicatie en samenwerking nog belangrijker dan het altijd al was in de IT. Een mooie vorm van communicatie is de 'Open Space'. Dit jaar voor het eerst op een TestNet evenement. Het idee voor de Open Space techniek is ontstaan nadat Harrison Owen constateerde dat deelnemers van conferenties de koffiepauzes vaak het interessantst vinden. Gedurende een Open Space sessie help je elkaar door je eigen ideeën, kennis en ervaring met de andere deelnemers te delen en samen tot een resultaat te komen.

In een Open Space sessie breng je zelf onderwerpen in en ga je van de ene naar de een andere sessie zodra je daar zelf zin in hebt. Net zoals je tijdens een koffiepauze zou doen!

Wat is een Open Space?

Tijdens een Open Space kom je met een groep mensen bij elkaar om samen een bepaald onderwerp of probleem te onderzoeken. Een Open Space vindt plaats zonder een agenda, tijdschema, voorzitter of notulist. Toch heeft iedereen aan het eind van een Open Space gezegd wat hij of zij wilde zeggen. Daarnaast heb je genetwerkt en heb je met iedereen ideeën uitgewisseld en samen nieuwe inzichten gekregen. De resultaten van een sessie leg je samen met de andere deelnemers vast op een groot papier of een collectie post-its. Na afloop van een sessie kunnen de deelnemers deze resultaten aan de muur hangen om ze zo met anderen te delen.

De Open Space techniek doet een beroep op het zelf organiserend vermogen van mensen. Je gaat naar de sessie die je het meest aanspreekt of start zelf een sessie. Omdat iedereen zelf voor een specifieke sessie heeft gekozen start bijna als vanzelf een levendige bijeenkomst waarbij mensen elkaar informeren, inspireren en aanvullen.

Spelregels van een Open Space

De spelregels van een Open Space Sessie bestaan uit vier eenvoudige principes en één wet:

De vier principes zijn:

- De deelnemers zijn de juiste personen
- Wat gebeurt, is het enige dat kan gebeuren
- Het begint wanneer het begint
- Als het voorbij is, is het voorbij

De wet van de twee voeten luidt als volgt:

Je bent verplicht om je twee voeten te gebruiken als je tijdens een sessie niets leert, niet geïnspireerd wordt of het je niet lukt om de sessie een betere richting te geven. Die twee voeten gebruik je om naar een andere sessie te gaan of de Open Space te verlaten. Deze wet heeft tot gevolg dat de minder productieve sessies snel eindigen.

Rollen in een Open Space Bijeenkomst

Gedurende een Open Space kan je verschillende rollen spelen. Daarbij bestaat geen goede of slechte rol. De rol die jij kiest is de juiste. Hieronder staan drie veel voorkomende rollen die je gedurende Open Space sessies kan spelen. →

1. De vlinder, die een sessie verlaat, wat in de ruimte rondfladdert en andere vlinders ontmoet en daar een informeel gesprek mee voert;
2. De bij, die een sessie verlaat, op zoek gaat naar een andere sessie en aan bestuiving doet: het overdragen van eerder opgedane kennis;
3. De trouwe hond, die de sessie uitzit totdat de groep een oplossing heeft geformuleerd.

Open Space Sessies op de Najaarsbijeenkomst

Iedere deelnemer krijgt de gelegenheid om een onderwerp voor een Open Space Sessie aan te dragen. Bij de ingang van de Najaarsbijeenkomst zie je het Open Space Sessie bord. Bij dit bord kan je om een post-it vragen waarop je een vraag kan schrijven. Deze vraag heeft betrekking op een onderwerp dat je graag in een sessie wil behandelen. De post-it plak je vervolgens op het Open Space sessiebord. Op die manier kunnen de andere deelnemers zien welke onderwerpen zijn aangedragen. Bij aanvang van een open sessie krijg je de gelegenheid om je onderwerp in een gloedvol betoog toe te lichten aan de andere deelnemers. Na deze betogen zullen de deelnemers de aangedragen onderwerpen prioriteren. De sessies starten vervolgens met de onderwerpen die de hoogste prioriteit hebben gekregen. De sessie eindigt wanneer de deelnemers vinden dat ze klaar zijn of als de tijd op is. ←

9,5 JAAR AGILE TESTEN IN NEDERLAND

Door Anko Tijman • anko.tijman@ordina.nl



Geweldig, een heel najaarsevenement rondom het thema Agile en Testen. Maar hoe zijn we als testers zo ver gekomen? En waar gaan we naar toe? Als early adaptor zal ik een tipje van de sluier oplichten!

Het begon allemaal met een presentatie op een TestNet thema-avond in maart 2003. Enkele weken daarvoor had ik Eveline Greveraers, destijds van de evenementencommissie, een mail gestuurd met de vraag of er niet iemand een presentatie over Agile kon geven.

Mijn toenmalige werkgever Van Meijel Automatisering was daar al een tijdje mee bezig en ik vroeg me af of meer mensen ervaring met Agile hadden. In een goed gesprek met haar (en ik dacht ook Rik?) op Utrecht CS besloten we dat ik de thema-avond maar moest verzorgen. Aldus geschiedde. Die avond kwamen de volgende onderwerpen aan bod:

- Wat is het Agile Manifesto?
- Wat is eXtreme Programming?
- Wat is Scrum?
- Wat is Crystal?
- Wat staat ons als testers te wachten?

Eurostar 2003

Bepaald nog geen antwoorden dus! Later dat jaar mocht ik dezelfde presentatie doen op Eurostar 2003 in Amsterdam en daar bleek dat de internationale test community Agile nog niet ontdekt had. Een wake-up call voor mij! Overigens was het wel zo dat op Eurostar2002 een sessie plaats heeft gevonden over testen in DSDM-projecten. Maar in die sessie was het woord Agile niet gevallen. Vreemd!

Bij mijn weten was de volgende thema-avond waar specifiek Agile aan bod kwam, in maart 2006. Een sessie die ik samen met Marc Evers van de Agile user Group XPNL heb gefaciliteerd. Met veel interactieve werkvormen was het een zeer energiegevend avond met veel leuke discussies! Het bleek wel dat Agile bij lang nog niet alle testers bekend was en dat de nieuwe denk- en werkwijze voor veel vraagtekens zorgde. Maar ook was duidelijk dat een groep testers bestond, vooral de pragmatici onder ons, die Agile zeker konden waarderen. De aanpak gaf richting aan samenwerken en teamfocus.

Eerste boekje

In maart 2007 kwam het eerste boekje over Agile testen van mijn hand. 'Testen als teamsport als antwoord op innovatie in ontwikkelprocessen' was de titel die qua lengte in geen verhouding stond met de dikte van het boekje. Centraal daarin kwamen (de) vier waarden te staan die een Agile testtraject onderscheiden van een traditioneel testtraject: Samenwerken, Eenvoud, Feedback en Flexibiliteit. Natuurlijk zijn die waarden niet exclusief voor Agile testen, maar ze zijn wel onontbeerlijk in een Agile context.

Internationaal waren de discussies over een eventuele rol van de tester in een Agile project wel geluid: die rol was er! Maar wat zou die rol dan moeten inhouden? Begin 2008 kreeg ik vanuit Ordina de opdracht om dan maar een boek over Agile Testen te schrijven. Dat heb ik samen met Eric Jimmink gedaan en in november 2008, →

tijdens Eurostar in Den Haag, werd het boek gepresenteerd. Nog geen maand later werd het boek 'Agile Testing – A Practical Guide for Testers and Agile Teams' van Lisa Crispin en Janet Gregory uitgebracht. Dat was internationaal de doorbraak van de Agile tester.

Bekender en breder

Vanaf 2009 werd het testen in Agile projecten langzamerhand echt bekender en breder toegepast. Daarvoor waren het vaak software huizen die Agile toepasten. Maar inmiddels gingen de eerste grotere bedrijven stiekem experimenteren. Als ik mij niet vergis, hebben we in augustus 2009 de eerste bijeenkomst van de werkgroep Agile Testen van TestNet gehouden, met daarin vooral mensen met ervaring. Dat waren in beginsel een stuk of zes mensen. Die groep is inmiddels heel wat groter. Op seminars, thema-avonden en evenementen kwam Agile steeds vaker in het programma voor. Daarnaast ontstonden internationaal heuse specialisaties, zoals Gojko Adzic met zijn Acceptance Test Driven Development. Het gebruik van specifieke tools voor testautomatisering zoals Selenium en FitNesse begon door te breken.

Focus veranderd

Het bleek wel dat Agile niet zonder meer geaccepteerd werd in traditionele organisaties. Vaak hadden testafdelingen jaren gevochten voor een eigen plekje. Was dat nu met Agile niet meer nodig? Maar inmiddels is Agile onder druk van noodzakelijke veranderingen in uitdagende tijden breed geaccepteerd. De Agile trend is onomkeerbaar gebleken. Ook in lineaire projecten zie je stand-ups die samenwerken faciliteren, worden projecten kleiner en dus eenvoudiger gehouden (als is het maar om budgettaire redenen) en zijn projectteams veel vaker multidisciplinair ingericht. De focus in het testen is daarmee veranderd van 'doen wat we horen te doen' naar 'doen wat we moeten doen om succesvol te zijn'. Een trend die ik van harte toejuich. Want een ding staat voor mij als een paal boven water: we doen onze projecten om waarde op te leveren voor onze klanten.

Innoveren

De trend die ik zie, is dat vanuit die multidisciplinaire teams het testen een veel breder draagvlak krijgt: ook architecten, ontwerpers en beheerders voeren testactiviteiten uit in het kader van het project. Wij zijn niet meer de enige tester, maar steeds meer de kennisdrager en facilitator. Daarnaast zal tooling steeds belangrijker en relevanter worden en zullen we nog meer gaan veranderen onder druk van trends als DevOps, Kanban en Agile Beheer. Wat mij betreft gaan we dus nog even door met innoveren! ←

IS EEN TESTMANAGER NOG NODIG BIJ AGILE?

Door Jeroen Mengerink • jeroen.mengerink@polteq.com •  @AngusVB



Met Agile werken verschuiven de verantwoordelijkheden steeds meer naar de uitvoerende teams. Bijvoorbeeld het plannen en schatten van het werk, wat voorheen door managers werd gedaan, wordt nu door de teams zelf gedaan. Blijft er nog wel genoeg werk over om de functie van testmanager te rechtvaardigen? Om dit te kunnen beoordelen moeten we eerst in kaart brengen wat de testmanager deed en wat daarvan overblijft in de Agile setting. Daarnaast moet gekeken worden naar nieuwe verantwoordelijkheden en taken die op de testmanager afkomen.

Wie doet wat?

De eerste taken die in gedachten komen wanneer ik aan de traditionele testmanager denk zijn:

- Testplannen maken en onderhouden;
- Testbeleid uitzetten;
- Productrisicoanalyse initiëren en uitvoeren;
- Inschattingen van het werk maken;
- Planning maken;
- Rapporteren over risico's, voortgang en kwaliteit;
- Managen van het testteam;
- Beheersen van het testproject.

Dit zijn voornamelijk organisatorische zaken betreffende het testen. Als we kijken naar wat er verandert bij Agile, zien we dat veel van deze taken verschuiven naar het Agile team.

(Test)plannen, testbeleid en rapportages

De Agile teams krijgen de verantwoordelijkheid om een inschatting te maken van het werk dat op ze af komt en dit werk vervolgens in te plannen. Een duidelijke indicatie dat er op dit gebied dus minder werk overblijft voor de testmanager. Wanneer de teams hun inschattingen maken in de planningssessies, moet er ook over risico's nagedacht worden om de hoeveelheid werk die in test gaat zitten, mee te kunnen laten wegen. Dus de productrisicoanalyse wordt opgesplitst per iteratie en door de teams uitgevoerd.

Dan hebben we de testplannen, het testbeleid en de rapportages nog. Eén van de Agile principes is dat werkende software boven uitgebreide documentatie gaat. Deze documentatie moet zo minimaal mogelijk gemaakt worden. Testplannen kunnen dus geen uitgebreide documenten worden! Bovendien is het handiger als de teams de vrijheid hebben om per iteratie te kijken wat er nodig is. Op het beleid kom ik later in dit artikel terug.

De rapportages liggen weer meer bij de Agile teams. Elke dag rapporteren de teams over hun status, wanneer ze in de daily stand-up laten weten wat ze gedaan hebben, wat ze gaan doen en wat de belemmeringen zijn. Meestal wordt hier ook naar de burndown chart gekeken – een grafiek die laat zien hoeveel uur werk er voor de huidige sprint openstaat. Dus de voortgang wordt bijgehouden en goed functionerende teams zullen bijsturen wanneer dit nodig is. →

De Agile testmanager

De Agile testmanager zal, onder andere door de bovenstaande verschuiving van taken bij Agile, meer op mensen gericht zijn en verantwoordelijkheid hebben om de juiste mensen aan de verschillende teams toe te wijzen. Hierbij moet de testmanager er zorg voor dragen dat voor Agile geschikte mensen worden aangenomen, dat de testers de mogelijkheid hebben tot het volgen van essentiële opleidingen en dat er voldoende groeimogelijkheden binnen de organisatie zijn. Niet elke goede tester past namelijk binnen Agile...

Het managen van het testteam is complexer geworden. Waar eerst het testteam samenwerkte, zijn de testers nu verdeeld over meerdere teams. In plaats van één bordje draaiend houden, moet nu een veelheid aan bordjes draaiend gehouden worden. Het is dus redelijk om aan te nemen dat aan dit aspect van testmanagement meer tijd besteed moet worden.



Een ander belangrijk aspect is het faciliteren van kennisdeling. Natuurlijk moet de kennis binnen het Agile team gedeeld worden, zodat het Agile team zo goed en efficiënt mogelijk kan werken. Vergeet echter niet dat wanneer er meerdere teams zijn, de testers uit de verschillende teams ook veel van elkaar kunnen leren. De testmanager kan hiervoor bijvoorbeeld elke twee à drie weken een testoverleg inplannen. Zo kunnen alle Agile teams voordeel halen uit de tegengekomen problemen en de best practices van de andere teams en blijft het volledige testteam ook leren.

Risicodenken

Het risicodenken binnen het team is erg belangrijk, maar dit moet ook op hogere lagen binnen een organisatie gebeuren. De testmanager kan bijvoorbeeld bij het maken van de product backlog en het opstellen van release planningen – ofwel iteratie overstijgende zaken – zorgen voor een risicoanalyse. Bij de releaseplanning moet de testmanager zorgen dat de impact van de verschillende backlog items en de verschillende teams op elkaar duidelijk worden. Over iteraties heen gekeken kan het ene item misschien alleen opgepakt worden wanneer een ander item afgerond is. Binnen een project met meerdere teams komt het regelmatig voor dat twee teams tegelijkertijd dezelfde delen van een systeem raken. Ook dit zal als risico tijdens releaseplanning en tijdens de planningssessies van de Agile teams moeten worden ingebracht. Ofwel er is een 'helicopter view' nodig op het project met de focus op kwaliteit en risico's.

Productkwaliteit

De business wil graag zicht kunnen houden op de productkwaliteit. Bij Agile testen hoeft niet alles opnieuw uitgevonden te worden, een gestructureerde aanpak voor het testen biedt nog steeds uitkomst. De Agile testmanager moet dus een generieke testaanpak maken waarbij over een aantal onderwerpen afspraken gemaakt worden. Denk hierbij aan de manier van bevindingen registreren, het omgaan met het testen van non-functionals en het omgaan met end-to-end testen. De bewegingsruimte die teams krijgen hangt af van een aantal factoren zoals de risico's, de volwassenheid van het testen, de volwassenheid van het team, enzovoort. Deze aanpak moet ruimte houden voor de Agile teams om zelf beslissingen te nemen. ➔



Conclusie

Bij kleine projecten met slechts één of twee teams is testmanagement waarschijnlijk geen fulltime werk en zal één van de meer ervaren testers uit de teams dit als rol kunnen invullen. Binnen organisaties met grote en/of veel projecten is voldoende werk voor een volwaardige Agile testmanagersfunctie. De rol van testmanager verdwijnt niet!

De focus van het werk ligt meer op people management. Daarnaast op het faciliteren en het controleren van de kwaliteit van het testen. Om het testen binnen een organisatie in goede banen te leiden zijn generieke richtlijnen nodig die door een testmanager opgesteld worden. Deze richtlijnen mogen de bewegingsvrijheid van de teams niet teveel beperken.

Het nadenken over risico's moet blijven gebeuren en niet alleen binnen de Agile teams, maar ook iteratie- en zelfs project overstijgend. De testmanager biedt een 'helicopter view' op projecten en kan zo op alle niveaus naar risico's kijken. Zeker bij meerdere parallel werkende Agile teams moeten risico's met betrekking tot onderlinge afhankelijkheden beheerst worden.

Dus: ja, ook in de Agile wereld is testmanagement nodig! ←

AGILE TESTEN VAN DATAWAREHOUSE & BUSINESS INTELLIGENCE

Door Iris Groenewoudt • iris.groenewoudt@ciber.nl en Armando Dörsek • armando.dorsek@sogeti.nl



Wij hebben het geluk te kunnen werken in een project dat aan alle kanten 'hot' is: werken als BI-consultant en testmanager voor een zorgverzekeraar in een grootschalig Agile Datawarehouse & Business Intelligence project. Wat zijn enkele markante verschillen tussen a) het testen van transactieverwerkende systemen en websites versus DW/BI omgevingen en b) tussen testen in waternal DW/BI-projecten en in Agile DW/BI-projecten

Wat is DW/BI?

Datawarehousing & Business Intelligence (DW/BI) is een specifieke tak van sport in de informatietechnologie. In een datawarehouse (DW) verzamelen we gegevens en leggen die consistent vast, zodat medewerkers op ieder gewenst moment over de benodigde gegevens kunnen beschikken voor besluitvorming en/of rapportage. De term Business Intelligence wordt daarbij doorgaans gebruikt voor de hulpmiddelen die deze informatie tonen, zoals: rapportages, dashboards, OLAP-tools enzovoort.

Begin 2012 heeft onze opdrachtgever, een grote Nederlandse zorgverzekeraar, een nieuw backoffice pakket geïmplementeerd. Om rapportages te maken, prognoses te doen en wettelijk verplichte bestandsaanleveringen maken is een Agile DW/BI-project gestart, dat daarin moet gaan voorzien.

In de loop van dit project constateren wij dat de karakteristieken van DW/BI-projecten en van de Agile werkwijze een duidelijk stempel drukken op de manier waarop getest wordt. We lichten hier de verschillen op twee assen toe, namelijk tussen:

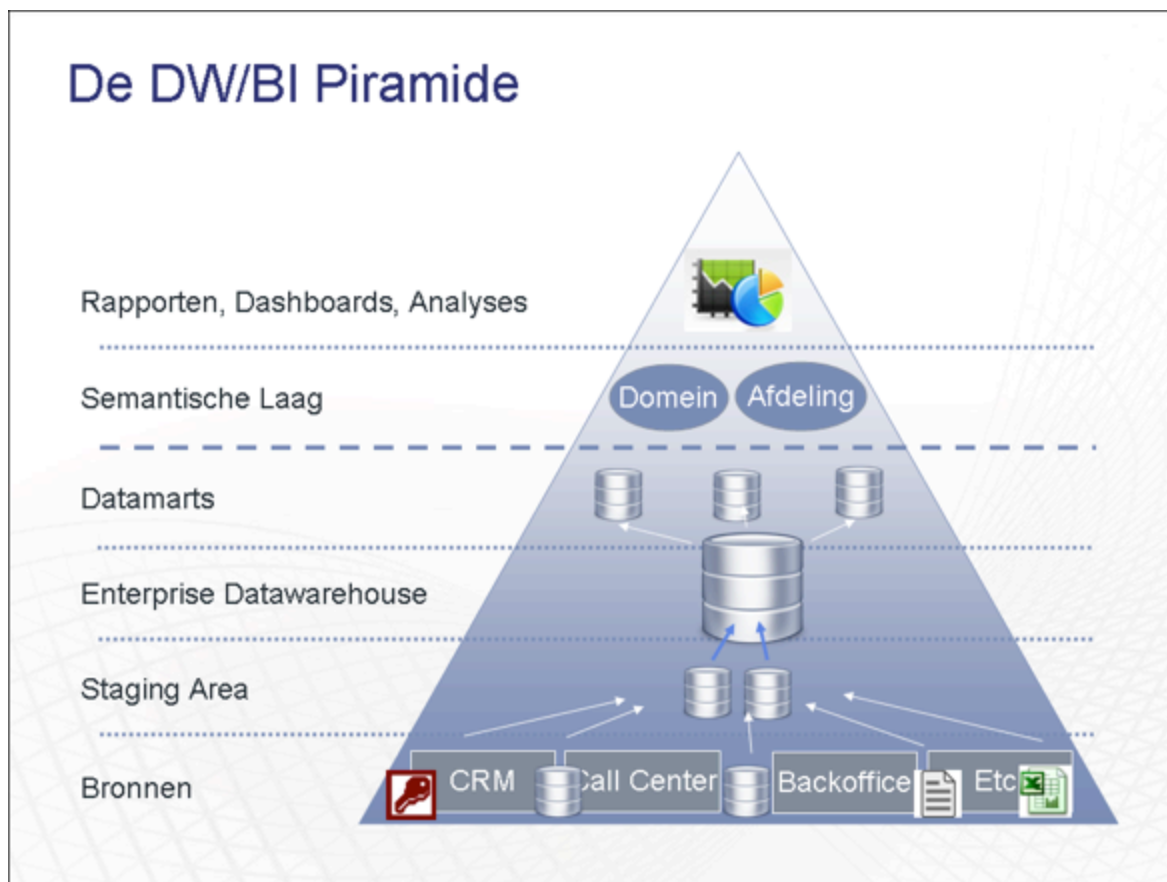
- A) het testen van DW/BI-omgevingen versus transactieverwerkende systemen en websites en
- B) het testen in Agile vs Waternal DW/BI trajecten.

BI/DW Waternal versus 'Overige' Waternal projecten

Hieronder presenteren we een standaard DW/BI-omgeving als piramide, waarmee in elk geval al het volgende duidelijk wordt:

- Vooral de componenten aan de top zijn uiteindelijk zichtbaar voor de gebruikersorganisatie. Eindgebruikers zijn vooral geïnteresseerd in deze BI-laag en veel minder in de 'dieper gelegen' onderdelen van het DW.
- Onder de spreekwoordelijke waterlijn ligt circa 80% van het systeem, zonder welke de genoemde rapportages enzovoort niet kunnen worden opgeleverd.
- Het is lastig om 'stukje bij beetje' een DW te ontwerpen en uit te breiden : als de fundamenteen niet goed zijn, leidt dit tot herstelwerkzaamheden. Doordachte architectuurkeuzes vanaf de start van het project zijn dus noodzakelijk.
- Voor het DW worden doorgaans veel verschillende bronnen ontsloten. Dat vereist afstemming met veel beheerders en stakeholders.
- Het DW is de 'eenduidige informatieomgeving'. De bronnen hebben elk hun eigen manier om met bedrijfsregels om te gaan. Met de toename van het aantal bronnen stijgt de hoeveelheid analysewerk. →

Uiteraard wordt in de literatuur en op forums ook gesproken over Agile BI. Vaak richt men zich daar niet op grootschalige DW/BI projecten maar op projecten met een beperkte scope (bijvoorbeeld een oplossing voor één afdeling) of alleen op het rapportage-deel.



Gebruikers van het DW komen uit verschillende afdelingen van de organisatie. Zij moeten overeenstemming bereiken wie er als eerste door het project geholpen wordt. Daartoe is bij onze klant een speciaal overleg ingericht, waarin medewerkers met een duidelijk mandaat zitting nemen.

Elk van de gebruikersgroepen kent verschillende acceptatiecriteria, afhankelijk van de doelstellingen van de informatieproducten. Voor de ene partij kan een dashboard ontwikkeld worden dat vooral een informerend karakter heeft, terwijl voor een andere partij een rapport wordt ontwikkeld dat opgenomen wordt in het jaarverslag. Deze situatie is vergelijkbaar met het testen van ERP-oplossingen, waarin de verschillende afdelingen ook vaak andere normen hanteren voor termen zoals 'juist', 'tijdig' en 'volledig'. Deze dienen voortdurend getoetst te worden per informatieproduct.

Definitiekwesties ('klant', 'verzekerde', 'prospect') moeten met de verschillende afdelingen afgestemd worden. Het DW is een 'single point of truth' voor de organisatie en het team moet vaststellen wat 'waar' is. Testers kunnen niet volstaan met de reacties van een acceptant op één specifiek rapport. Zij hebben een belangrijke signaalfunctie naar analisten, architecten en zogeheten data stewards. Bij de klantorganisatie is een zogenoemd bedrijfsbreed gegevenswoordenboek in opbouw.

Testen van ETL-software

ETL-software (Extractie, Transformatie en Laden) bestaat uit het oppakken van gegevens, integreren, bewerken en vervolgens in het DW laden. Het testen van ETL-software is een omvangrijk onderdeel van het DW/BI-project. →

Daartoe is vaak bovengemiddelde kennis van SQL en ontwikkeltools vereist. Bij het testen van bijvoorbeeld een website is diepgaande kennis van de onderliggende lagen (database-, web- en applicatielagen) vaak een luxe, geen harde noodzaak. Bovendien zijn daar meer testtools 'off the shelf' beschikbaar dan bij DW/BI-projecten. Bij het testen van het DW is deze kennis een randvoorwaarde.

Omdat de zichtbare eindproducten pas laat in het traject worden opgeleverd, kan pas laat gestart worden met een gebruikersacceptatietest (GAT). Het is dus belangrijk om maatregelen te nemen die het ontstaan van fouten in de dieper gelegen lagen voorkomen. Herstelwerkzaamheden onder de waterlijn hebben impact op veel eindproducten en zouden kunnen leiden tot veel hertesten en regressietesten.

Een DW richt zich op bedrijfscijfers en wordt door het management gebruikt voor belangrijke beslissingen en voor financiële rapportages. Hierdoor is vaak de interne Audit-afdeling een belangrijke partij om rekening mee te houden. Testers zijn gelukkig steeds vaker een belangrijke voortrekker van de afstemming met de Audit-afdeling en door gedeelde methoden, technieken en modellen vaak een goede gesprekspartner.

De kwaliteit van de gegevens in de bronsystemen is rechtstreeks van invloed op de kwaliteit van de gegevens in het DWH. We adviseren om een tester en business analist samen te laten kijken naar de datakwaliteit in een nieuw aan te sluiten bronsysteem. Voor de tester is deze informatie ook van belang bij het toetsen van de robuustheid van de transformatieregels. Een lagere datakwaliteit in de bron betekent doorgaans dat robuustere (en complexere) transformatieregels nodig zijn. En dat betekent méér testactiviteiten.

BI/DW Waterval versus BI/DW Agile projecten

Hulpmiddelen en testautomatisering

De testautomatisering heeft in het voorbeeldproject een boost gekregen als gevolg van de Agile/Scrum aanpak. Doordat 'het team' verantwoordelijk is voor de op te leveren producten, niet individuen, werkt dit samenwerken in de hand. Dit stimuleert testautomatisering en het ontwikkelen van overige hulpmiddelen zoals checklists.

De testers in het team hebben aan het begin van het project uitgesproken wat zij wilden testen en vooral: waarom. Dit leidde tot begrip bij de ontwikkelaars in het team en zij hebben een geautomatiseerde oplossing gerealiseerd, die alle teamleden elke sprint weer werk bespaart.

Anders dan bij een watervalproject kun je niet volstaan met een eenmalige controle op het naleven van standaarden en naamgevingconventies aan het einde van het project. Elke sprint levert het team bruikbare software op – dus zul je die software daarop willen toetsen. Hiervoor ontwikkelde het team een controlemechanisme, een programma dat de code controleert op het naleven van de standaarden.

Een tweede oplossing betreft audit functionaliteit: een mechanisme waarmee de tester kan vaststellen dat de data volledig overkomt, van de ene naar de andere laag in het DW. Dit vervangt veel handmatige acties van de tester en is bovendien uitgebreid om ook voor audit doeleinden gebruikt te kunnen worden door beheerders en auditors (zie vorige paragraaf). Het inbouwen van dergelijke functies voor audit- en beheerdoeleinden zijn een mooi voorbeeld van de verschuiving van Quality Assurance naar Operational Assurance

Het Agile Manifesto geeft aan dat werkende software gaat boven documentatie. In de Agile aanpak is eventuele documentatie waarschijnlijk pas tegelijk met de software gereed. Vooraf uitgebreide testscripts opstellen op basis van ontwerpdocumentatie is daardoor vrijwel onmogelijk. In plaats daarvan zijn de Unittesten uitgebreid, wordt méér met checklists en tooling van ontwikkelaars gewerkt en wordt uiteraard de Product Owner zo vroeg mogelijk in het testtraject meegenomen. →

- Het team is verantwoordelijk voor het bereiken van de 'Definition of Done', de teamleden werken samen aan het realiseren van klantwaarde. Testers worden vanaf de start betrokken en dat past volledig in de gedachte van PointZERO®, waarin men ook spreekt over 'Quality and Collaboration'. Agile werken vergt een moedige, open en eerlijke houding van alle teamleden. De tester in een Agile traject zal weinig als een schooljuf fouten rapporteren, maar vooral een coachende rol aannemen. Wij herkennen ons in de tekst van Ian Ross (ANZTB): '... and therefore a tester is not there so much to tell the developer he has got it wrong, but you're a second pair of hands to make sure you're doing it right'.

Product Risico Analyse en de teststrategie

Bij de start van een 'waterval DW/BI-project' zal de testmanager een Product Risico Analyse (PRA) uitvoeren op basis van de (complete en statische) requirements. Hij gaat uit van een specifieke methodiek/technologie gecombineerd met een inschatting van de risico's van dat moment. Wat is de meest kritieke rapportage? Welke laag is het meest vatbaar voor fouten? Hoeveel ervaring hebben we met dit domein? De resultaten van de PRA zullen leiden tot een testaanpak met afspraken over testdiepgang, wie doet wat, met welke testspecificatietechnieken, 'quality gates' enzovoort. Er wordt gedacht vanuit de keten: belangrijke of risicovolle ketens testen we eerder en uitgebreider dan de andere. Als in de loop van het traject aanpassingen nodig zijn op de scope of de volgorde waarin componenten worden opgeleverd, leidt dit er vaak toe dat er minder wordt getest, of met minder diepgang. Bij de start van een Agile traject is uiteraard ook de scope bekend. Deze heeft echter de vorm van een Product Backlog waarop de, door de klant geprioriteerde, User Stories staan. We schatten de benodigde inspanning in per User Story en vullen daarmee de zogeheten Sprint Backlog. In korte cycli van circa twee tot vier weken (Sprints) wordt gewerkt aan de 'potentially shippable items'. Eén van de grote voordelen voor de klant: als het project (vroegtijdig) beëindigd wordt, dan is met de afgeronde sprints toch nog waarde toegevoegd in de vorm van bruikbare software. In de watervalvariant zou de klant dan mogelijk vooral documentatie ontvangen en geen of slechts enkele werkende rapporten – en mogelijk niet eens de meest belangrijke.

Test Planning

De grote kracht van Agile ligt onder andere in zeer korte 'feedback loops', zowel tussen projectleden onderling als tussen het projectteam en de rest van de gebruikersorganisatie. Daarbij kan blijken dat een bepaald informatie-product bij nader inzien tóch meer prioriteit moet krijgen – en dat daardoor eerder aan een andere keten moet worden gewerkt dan verwacht. Het is waarschijnlijk dat hierdoor de volgorde van het ontwikkelen (en testen) anders zal uitpakken. Om bij de aanvang van het project een gedetailleerde teststrategie en -planning op te stellen is dan niet waardevol. In Agile trajecten zal meer behoefte zijn aan een testaanpak op hoofdlijnen, een set basisafspraken voor de Definition of Done. Per Sprint en per User Story zal vooral situationeel worden bepaald welke aanvullende testactiviteiten nodig zijn. Als bij de Sprint Planning rekening gehouden moet worden met 'extra' testinspanning, dan kent het team meer Story Points toe aan de User Story.

Technical Debt

Het werken in sprints stimuleert teamleden om zich te richten op wat nú belangrijk is. Refactoring is nodig om ook aan de langetermijndoelstellingen te kunnen voldoen. De Product Owner kan vaak niet overzien wat de technische consequenties zijn van het uitstellen van refactoring. In elke sprint reserveert ons team een aantal Story Points om zogeheten 'Technical Debt' te voorkomen. →

Ook het opzetten van een (geautomatiseerde) regressietestset leent zich ervoor om vanaf de start van het project als 'Technical Story' op te nemen in de product backlog.

Valkuil: Uitstellen van Regressietesten

In iedere sprint levert het team nieuwe 'potentially shippable items' op. Nieuwe en aangepaste software mag de eerder gerealiseerde software niet nadelig beïnvloeden en dat kan aangetoond worden met een regressietest. Na een aantal sprints levert het team de uitbreidingen in steeds hoger tempo op. Refactoring op eerder ontwikkelde componenten is steeds vaker aan de orde. Het team is groot en de individuele ontwikkelaars kunnen slechts met moeite voorkomen dat zij elkaars software raken. Ook de hoeveelheid data in de bron en in het DW neemt exponentieel toe.

We constateren nu dat het opstellen van regressietesten (al dan niet geautomatiseerd) onvoldoende aandacht heeft gehad. De regressietestset ontbrak als 'Technical Story' op de Product Backlog. Er zijn weliswaar scripts voorhanden maar deze hebben nog niet de samenhang die van een regressietestset verwacht mag worden. Het opzetten van een regressietestset zal nog een behoorlijke inspanning vergen.

Tegelijk is de omvang van het systeem nu zodanig gegroeid dat een subsetting mechanisme noodzakelijk wordt om onder andere de laadtijden binnen de perken te houden. Niet alleen voor de Regressietest maar ook voor Unit- en Systeemtesten.

We adviseren daarom, uit ervaring, om de regressietesten (en subsetting mechanismes) al bij aanvang van het project expliciet op de product backlog te plaatsen respectievelijk om deze in de Definition of Done op te nemen.

Samenvatting

Het bovenstaande kunnen we als volgt samenvatten:

- De natuur van DW/BI-projecten maakt het noodzakelijk om vanaf de start vast te stellen dat de juiste dingen worden gedaan op een goede manier. Vooral in een Agile omgeving krijgt de tester een kans om daar een actieve rol in te spelen. Onder andere de gewenste 'verschuiving naar links in het V-model' krijgt daarmee een zet in de goede richting.
- In Agile teams leveren testers duidelijker ondersteuning en coaching aan teamleden, in plaats van 'puur kritiek'. Het toevoegen van waarde staat centraal.
- Agile teams kunnen samen, eerder dan in een waterval model, besluiten dat het voor hen zinvol is om ondersteunende tooling te realiseren, bijvoorbeeld gericht op testen en auditing.
- Leden van Agile teams pakken urgente zaken op, niet alleen datgene waar zij (nu al) goed in zijn. Dit biedt hen kansen om te leren en de kans is groot dat testers de mogelijkheid krijgen om bij te dragen aan andere onderdelen van het project dan testen.
- Testers in DW/BI-projecten moeten veel overleggen met diverse stakeholders. Dat vereist goede communicatieve vaardigheden, technisch inzicht en gevoel voor wat belangrijk is voor gebruikers, beheerders en bijvoorbeeld de architect of auditor. →

- Bij een Agile aanpak wordt relatief snel de eerste bruikbare software opgeleverd. De hoeveelheid data op het (productie) datawarehouse zal snel groeien. Als niet vanaf de start van het project wordt gewerkt aan (geautomatiseerde) regressietesten en een subsetting mechanisme, dan worden de doorlooptijden van het testen mogelijk snel onhoudbaar.
- Het testen in DW/BI-projecten vereist relatief veel technische kennis, dit maakt het 'matchen' van testers op DW/BI-projecten lastig.
- In DW/BI-projecten wordt nogal eens ervoor gekozen om een ontwikkelaar te laten testen – in dat geval is een coachende tester waardevol.

Referenties

- 'Hoe Testers en Auditors elkaar vinden - Zodat u beter slaapt', Egbert Bouman (Valori), 27-04-2011. <http://www.xr-magazine.nl/artikelen/961/testen/ho-testers-en-auditors-elkaar-vinden-zodat-u-beter-slaapt>
- Linda Hayes, 'You're Either On The Train or On the Tracks: Radical Realities Shaping Our Future', keynote STAR WEST 2010.
- 'PointZERO: The First Time Right Vision', Rik Marselis, 21-6-2012 (Keynote TestEXPO2012). <http://www.tmap.net/sites/tmap.net/files/attachments/PointZERO%20the%20first%20time%20right%20vision%20TestEXPO%20v1.pdf>
- 'Van data tot informatie; het testen van Business Intelligence', Bart Vrenegoor (Sogeti), 18-12-2012, Presentatie op 'Dutch Testing Conference 2012'. http://www.tmap.net/sites/tmap.net/files/attachments/Van%20data%20tot%20informatie%3B%20het%20testen%20van%20Business%20Intelligence_0.pdf ←

UIT HET OOG, UIT HET HART?

Door Yvonne van Zaanen • yvonne.vanzaanen@triodos.nl



Drie jaar geleden is de ICT-ontwikkelfdeling van de Triodos Bank overgegaan van de watervalmethode naar de Scrum-methodiek. Wij testers werden hierbij uit onze comfortabele positie achter de muur, met als houvast het V-model en als schild de testmanager, ineens midden tussen de ontwikkelaars geplaatst. En in plaats van regelmatige discussies over testtechnieken werden wij geconfronteerd met codetaal en ontwikkelaarshumor. Waren we onze testmaatjes kwijt? Of konden we eindelijk helemaal los en lekker onze eigen gang gaan, ingegraven in het Scrum-team? Had dit ook zijn keerzijde?

Pilotproject

Na een lange zoektocht naar methoden en tools om nu eens écht het ontwikkelproces te verbeteren en de klant beter te kunnen bedienen, was in 2009 hét moment aangebroken; Scrum werd geïntroduceerd bij de Triodos Bank. Met behulp van een externe partij startte in mei 2009 een pilotproject met een Scrum-team. De rest van de afdeling bleef nog even watervallen tot de pilotgroep de eerste kinderziekten had geëlimineerd. Dit leek onder controle halverwege 2010, waarna besloten werd de hele ICT-ontwikkelfdeling aan de Scrum-methodiek te onderwerpen. Er werd een aantal Scrum-teams geformeerd, bestaande uit drie tot vier ontwikkelaars, één tot twee informatieanalisten en één tester. Voor de testers betekende het dat het 'testteam' in de tot dan toe bestaande vorm werd opgeheven. Dit uitte zich in eerste instantie door de fysieke verhuizing en de verspreiding van de testers over de Scrum-teams.



Teamleider

De rol van testmanager was ondertussen vervangen door die van teamleider. In deze veranderde functie was de teamleider niet meer verantwoordelijk voor het operationeel aansturen van de testers. De teamleider bleef wel in stelling voor de HR-gerelateerde zaken en het afdeling/bedrijfsbreed meedenken over en bepalen van de strategie betreffende (de rol van) het testen.

Het werk werd verdeeld over de Scrum-teams, die vervolgens ook als team gezamenlijk verantwoordelijk werden voor de oplevering van de software; op tijd en van goede kwaliteit. En toen gingen we los. Het hele scala van Scrum-gerelateerde zaken passeerde de revue: backlogs werden gevuld, prioriteiten van user stories bepaald (door de product owners), poker- en planningssessies gehouden, sprints gevuld, stand-ups ingevoerd, retrospectives ingepland. →

Weerstand

Scrum was voor iedereen op de afdeling. De weerstand die de ontwikkelaars van nature hebben tegen testen en zijn vertegenwoordigers, de testers, werd in het team en tijdens de stand-ups niet onder stoelen of banken gestoken. Een aantal testers pasten zich als van nature gemakkelijk aan de veranderde omstandigheden aan, voor anderen ging het wat moeizamer. Werd testen in de watervaltrajecten gezien en afgedaan als een noodzakelijk kwaad, nu moesten de testers nog prominenter laten zien - en een ieder individueel - dat testen een noodzakelijk goed is, ieders verantwoordelijkheid en dat niemand binnen het Scrum-team er aan ontkomt. Dat vraagt van de testers een iets andere attitude dan men gewend was in de 'oude' situatie. De tester wordt geacht zich nog meer te profileren als 'testgeweten' van het team (hoewel sommige testers zich niet veel bij deze term konden en kunnen voorstellen) en de toegevoegde waarde van zijn werk duidelijk te maken aan de andere teamleden. Nee, geen indoctrinatie, communicatie was het devies!

Enige schroom

Aangezien er verwacht wordt dat je in het multidisciplinaire Scrum-team ook multidisciplinaire taken uitvoert, zou je ervan uit kunnen gaan dat ontwikkelaars en informatieanalisten ook testtaken oppakken. Dit werd echter niet zo vanzelfsprekend ingevuld. Er was enige schroom bij 'niet-testers' de testtaken op te pakken en vanuit de tester enige achterdocht als een ander teamlid met het testen aan de haal ging. Ook hier ging wat tijd overheen, voordat de ontwikkelaars, informatieanalisten en tester als team de testonderdelen gezamenlijk gingen oppakken. Aangezien de niet-testers, geen tot weinig testervaring hadden werd het een taak van de tester om het testenthousiasme over te dragen en daarbij coachende vaardigheden ten toon te spreiden.



Bij Scrum kan de tester niet een afzonderlijk en eigen testfeestje organiseren en bouwen. Het voorbereiden, uitvoeren en afronden van de tests wordt gedeeld met iedereen van alle betrokken disciplines.

Samenwerking met de Scrum-teamleden werd ook geïntensiveerd door samen testscenario's te maken en vervolgens met de ontwikkelaars te bespreken welke testen wanneer (unit testen, systeem testen) en door wie uitgevoerd zouden worden.

Een plek veroveren

Een van de gevolgen van alle veranderingen was dat de testers zich voornamelijk richtten op het Scrum-team om daar een plek te veroveren. De tester voelde zich al snel meer verbonden met het Scrum-team dan met het testteam. Het 'testteam-gevoel' was op een gegeven moment helemaal weg. Nieuwe testers die binnenkwamen werden in een Scrum-team geplaatst en wisten in eerste instantie en gedurende de eerste periode niet eens wie in de andere teams de tester was. Testers hadden nauwelijks nog contact met elkaar. Er was nog wel een testteammeeting maar men wist eigenlijk niet goed van elkaar waar men mee bezig was. Samenhang ontbrak, iedereen gebruikte ook eigen templates en dergelijke. De Scrum-methodiek laat erg veel vrijheid over voor testen en testers. De documentatie vertelt over waar je mee geconfronteerd wordt als tester als besloten is Scrum te gaan werken, wat Scrum is en hoe je je moet aanpassen, maar als er over 'team' wordt gesproken is dat het Scrum-team en niet het testteam. De vraag kwam op of het eigenlijk nog wel nodig is dat je je als tester ook 'verbonden' voelt met de andere testers of dat de focus vooral ligt bij de collega's in het Scrum-team. →



Echter, na de Scrum-gewenningsperiode bleek dat de testers het contact en het sparren met testcollega's gingen missen.

Als een van de eerste stappen om elkaar weer eens te zien en meer van elkaar te weten te komen, behalve de gebruikelijke werkzaken, werd de wekelijkse testteam-lunch geïntroduceerd. Was in de oude situatie de tweewekelijkse testteam-meeting vooral eenrichtingsverkeer met mededelingen van de testmanager met eventueel een rondje langs de velden, nu zijn de testers zelf verantwoordelijk voor een deel van de invulling van deze meetings.

Elkaar afwisselend worden er presentaties gehouden over de projecten die men test in het Scrum-team of onderwerpen aangesneden die ons allen aangaan. Zo is de kennisdeling en -overdracht weer nieuw leven ingeblazen.



Tijdens de meetings discussiëren testers over de gehanteerde aanpak en wisselen tips en trucs uit over hoe nog beter te kunnen testen.

Ook is er wekelijks een paar uur ingeruimd voor pair-testing om bij elkaars werk betrokken te blijven.

Al met al zijn we er goed in geslaagd om onze plek te vinden binnen de Scrum-teams, waarbij we ons ook onafhankelijk/kritisch kunnen opstellen. Dit is onder andere een gevolg van het opnieuw tot leven brengen van het testteam-gevoel. ←

TESTEN OP 81 VIERKANTE METER

Door Ard Kramer • a.kramer@eclipseit.nl • [@ard_kramer](https://twitter.com/ard_kramer)



Een volleybalteam werkt op een klein veld van 81 vierkante meter keihard om de overwinning te behalen. Dit stelt hoge eisen aan samenwerking, coaching en strategie. Het is interessant om na te gaan wat wij als testers, actief in een Scrum-team, kunnen leren van zo'n volleybalteam. En je weet het, je ziet het pas als je het door hebt.

Wie herinnert zich niet hét sportmoment van de twintigste eeuw? Zes lange volleybalmannen die hun ultieme droom bereiken op de Olympische Spelen van Atlanta.

Na jaren van training en verloren finales op momenten dat het er op aan kwam, op het moment suprême winnen met een minimaal verschil van de grootste concurrent uit die tijd, de Italianen. Een geweldig moment!

Maar wat heeft zo'n volleybalwedstrijd nu met Agile te maken? Veel! Om te beginnen gaat het bij Agile net als bij volleybal om in een klein team en onder hoge druk te presteren. Bij volleybal is er letterlijk sprake van een klein veld, 81 vierkante meter. Maar ook bij een Scrum-team werken de spelers dicht bij elkaar aan het resultaat.



Primaire emoties

Ik vind sport in het algemeen interessant, omdat te zien is hoe we als mensen handelen als we dicht bij onze primaire emoties komen. Je kunt je (emoties) niet verbergen op een sportveld (soms zien we daar ook de uitwassen van, vooral op het voetbalveld) en we laten ons dan helemaal gaan. Maar het betekent wel dat je moet leren hoe je met elkaar in die omstandigheden omgaat. En hoe je al die adrenaline en emoties in goede banen leidt om tot een ultieme prestatie te komen.

Je moet als team gaan voor de ultieme prestatie. En dat vergt nogal wat van de teamspirit. Een team bestaat uit individuen die allemaal hun eigen ego en temperament hebben. Een uitdaging waar ik als volleybalcoach regelmatig mee te maken heb gehad, met goede en slechte ervaringen. Wijze lessen waar ik later op terugkom.

Geen compromis

Een tweede punt dat sport interessant maakt: sporters verdragen geen compromis. Het is heel simpel, je bent een winnaar of een verliezer. En verliezen ... moet je leren. Uit tv-interviews na afloop van een wedstrijd blijkt vaak genoeg hoe moeilijk dit is. Als verliezer heb je gewoon verloren. Kniezen helpt niet. Marc Lammers (hockeycoach) stelt het duidelijk: 'Verliezers hebben een excuus, winnaars een plan'. Bij Agile gaat het ook om winnen of verliezen. Als je niet in staat bent om op de deadline werkende software op te leveren, dan is het simpel: je hebt verloren. →

Een plan

Het laatste punt met betrekking tot sport dat ik naar voren wil brengen heeft te maken met een andere uitspraak van Marc Lammers: 'De ultieme prestatie vereist een plan'. Wil je winnen op de Olympische spelen, dan vereist dat een vierjarenplan. Hierbij zie je dat werkelijk alles uit de kast wordt gehaald om het doel te halen. Helaas zijn hier ook negatieve voorbeelden van, zoals het gebruik van doping. Maar gelukkig zijn er ook vele goede voorbeelden, waarbij fundamenteel onderzoek is verricht en waarbij talrijke innovaties voor de sport zijn ingevoerd. Voorbeelden van nieuwe ontwikkelingen zijn de toepassing van de toegenomen kennis over (groeps)processen en het inzetten van visualisaties om prestaties te verbeteren. Een leuk voorbeeld komt van Marc Lammers die ontdekte dat koude baden het herstel van zijn speelsters bevorderde. De Argentijnen zagen dit ook en werden nieuwsgierig. Marc liet een speelster van zijn team bewust verkeerde informatie over deze aanpak naar hen lekken. De Argentijnse trainersstaf kreeg op deze manier de informatie doorgespeeld dat je wel twee uur in zo'n koud bad moet liggen om goed te herstellen. Dat was veel te lang, waardoor ze na afloop nog uren rillend rondliepen. Dit tot grote hilariteit van het Nederlandse team. Wat kun je hiervan leren? Als je nieuwe ontwikkelingen ziet, let dan goed op wat de gedachte hierachter is, anders loop je in een valkuil die je 'vele rillingen' kan opleveren.

Ik wil vanuit mijn ervaring als volleybalcoach drie aspecten de revue laten passeren om jullie aan het denken te zetten over je rol als tester en 'speler' in een Scrum-team of Agile omgeving.

A. Teamplay

1. Afspraken maken over samenwerking

Teamplay in volleybal is ontzettend belangrijk. Als de bal aan jouw kant is, ben je volledig in control en heb je geen last van een tegenstander. Maar je moet wel nauw samenwerken om de bal 'klaar te leggen' voor de ultieme aanval om dat hele belangrijke punt te maken. Iedereen van het team is daarbij belangrijk. Deze grote afhankelijkheid maakt dat er in je team afspraken over 'het proces' zijn gemaakt om optimaal samen te werken. Het gaat om details. Als de speler bij het net een stapje naar rechts doet, heeft dit gevolgen voor de speler in het achterveld. Vooraf maak je hier generieke afspraken over en deze worden tot in den treure getraind.

Van die sterke afhankelijkheid is ook sprake in een Scrum-team: als een informatieanalist iets bedenkt, heeft dat gevolgen voor de bouwactiviteiten van de programmeur en voor het testwerk van de tester. Deze grote mate van afhankelijkheid wordt af en toe zwaar op de proef gesteld. Dit is een van de redenen dat volleyballers na het maken van een punt altijd even gezellig bij elkaar komen in het midden. Daar stimuleren de spelers elkaar, maar wordt ook kort informatie uitgewisseld over de praktijk. Bijvoorbeeld wat doet een tegenstander in een bepaalde situatie en hoe gaan we daar op reageren? Het wordt besproken in een echte Scrum!

In de praktijk heb ik geleerd dat de wijze waarop de teamleden elkaar aanmoedigen en informatie aan elkaar doorgeven van groot belang is. Je hebt daarbij te maken met heel uiteenlopende persoonlijkheden. Om dit proces van feedback en informatie-uitwisseling te stroomlijnen heb ik een keer de hulp ingeroepen van een sportpsycholoog. Met de psycholoog zijn we aan de hand van een aantal testen met elkaar gaan verkennen waar onze sterke en zwakke punten liggen. →

En we bespraken vragen als: hoe wil je als speler aangesproken worden? Moet je juist aangemoedigd worden of wil je juist met rust worden gelaten als het spannend wordt of als er fouten gemaakt worden? Hierover maakten we gezamenlijke afspraken en daar op vielen we terug in het heetst van de strijd.

Worden dergelijke afspraken in Scrum teams ook vooraf gemaakt? Bijvoorbeeld hoe je aangesproken wil worden? Mag het hard en direct of heb je liever dat je op een rustig moment in een rustige omgeving feedback ontvangt?

2. Een evenwichtig team

Een ander belangrijk aspect van het teamplay is het feit dat één sterspeler ontoereikend is. Je mag niet vertrouwen dat één goede speler het hele spel kan maken en dragen. Tenslotte geeft dit een te grote belasting van zo'n speler, die dan ook niet meer de kwaliteit kan bieden die nodig is voor een goede prestatie. Je zult een evenwichtig team moeten creëren waar mensen elkaar aanvullen en vooral niet hetzelfde zijn. Zo heb ik de waarde van de zogeheten waterdrager leren kennen. In het team waarmee ik een keer Nederlands kampioen werd zat een speler die niemand opviel. Hij had echter een belangrijke waarde: hij scoorde weliswaar niet veel maar hij maakte geen fouten waardoor we geen kostbare punten verloren. In een team is het belangrijk dat iedereen van waarde is. Dan pas ben je samen sterker.



3. Continu willen verbeteren

De laatste belangrijke les in het teamplay, dat geldt voor volleybal is het motto 'speel de bal altijd beter dan je voorganger'. De gedachte hierachter is dat spelers nogal eens het hoofd laten hangen als er een slechte eerste pass wordt gegeven: 'De rally zal wel niets meer worden na zo'n slechte eerste bal'. Dit is een funeste gedachte om een goed resultaat te behalen. Als je het omdraait en je probeert elke bal beter te spelen dan ben je bereid om fouten of onvolkomenheden te accepteren en te verbeteren.

Als dit in je Agile team gebeurt zal het product bij elke stap die je maakt beter worden. De 'state of mind' van het team is dus van belang: positief denken en je realiseren dat je het product weer beter moet afleveren dan je het ontvangen hebt. Als tester moet het je uitdaging zijn om het product voorwaarts te helpen. Niet tevreden zijn als je de grootste fouten er wel ongeveer uit hebt gehaald maar doorgaan om werkelijk van meerwaarde te zijn voor een goed eindresultaat.

B. Fouten maken en zelfcoachend team

Het tweede aspect gaat in op het coachen van elkaar binnen het team. Als coach sta je naast het veld buitenspel en moet je vertrouwen op je team; dat de taken worden uitgevoerd zoals je vooraf samen hebt bedacht. Hierbij geldt bij het volleybalspel dat als je een fout maakt dit een punt kost. Verlies je daar een wedstrijd mee? Nee, duidelijk niet, sterker nog fouten maken horen er bij en het gaat er dan om wat voor soort fouten je maakt. Maak je ze omdat je iets uitprobeert omdat je voor een probleem gesteld bent dan kan dit een positieve fout zijn. Het gaat om de intentie. Hierbij is het wel belangrijk om binnen je mogelijkheden te blijven. Een tester moet de ruimte hebben om een andere aanpak te proberen, om op nog beter wijze fouten uit het systeem te halen. Het mag dan niet iets zijn wat de tester nog nooit heeft gedaan. Je moet natuurlijk 'geen nieuwe dans leren de dag voor het bal'. Is het echter een techniek waarmee uiteindelijk toch niet het resultaat wordt bereikt, →

dan kun je dit een positieve fout noemen. Voor het grotere goed, het resultaat, heb je een weloverwogen risico genomen. Hierbij is het belangrijk dat teamleden elkaar hierbij scherp houden, en aangeven wanneer je risico kunt nemen. Niet een dag voordat de presentatie van je software komt maar bij het begin van de sprint, dan is er nog ruimte. Geef hierbij elkaar ook ruimte, houd rekening met elkaar en help elkaar. Als een speler op een bepaalde manier aangespeeld wil worden, dan probeer je hier zo goed mogelijk aan te voldoen. Zo kan een ontwikkelaar goed rekening houden met hoe de tester aan de gang kan gaan. Hij moet niet per se zijn maniertjes willen door-drukken omdat hij hiermee makkelijker programmeert maar denken in het grotere geheel: 'Als ik een aanpassing maak in mijn werkwijze, dan kan de tester beter zijn werk doen' Door dit met elkaar te overleggen coach je elkaar tot een beter resultaat. Als dit soort processen door de teamleden goed worden opgepakt, dan zie je gedurende de wedstrijd spelers het initiatief nemen en samen oplossingen bedenken.

C. Een goed resultaat bereiken

Het derde aspect gaat over winnen, het uitvoeren van een succesvolle sprint. Hoe doe je dat? Als volleybalcoach vind ik het resultaat totaal onbelangrijk. Wat? Resultaat onbelangrijk? Hoe kun je dat zeggen? Ja, dat durf ik te zeggen. Het gaat mij namelijk om de weg die je kiest en het perspectief dat je hebt.

Een goed resultaat is het gevolg van de goede weg kiezen

Om te beginnen met de weg die je kiest: Ik weet dat als de spelers maximaal samenwerken, hun taken uitvoeren zoals je hebt afgesproken, elkaar aanspreken en inventief zijn om problemen op te lossen het vanzelf wel goed komt met het resultaat. Het proces is belangrijker dan het resultaat.

Door elkaar feedback te geven op onderdelen die goed moeten gaan, dan zal het eindresultaat goed komen. Als coach geef ik vooraf bij een wedstrijd een aantal uitdagende tussendoelen mee, die de spelers geheel in eigen hand hebben. Dit kan per speler verschillen. Een speler die serveren lastig vindt, krijgt als opdracht, ten minste drie keer achter elkaar goed serveren.

Het voordeel van zo'n opdracht is dat, ook al verliest het team de wedstrijd, hij nog steeds positieve feedback kan krijgen omdat hij zijn opdracht goed heeft uitgevoerd. Voor een tester kan zo'n uitdaging liggen in het vinden van een aantal fouten in cruciale onderdelen software. Dat uiteindelijk de opdrachtgever ontevreden is omdat hij toch de verkeerde user stories had opgesteld mag de tester er niet van weerhouden toch goed werk en een goed gevoel aan de sprint over te houden.

Een goed resultaat is het gevolg van een goed gekozen perspectief

Het perspectief dat je kiest is heel belangrijk voor het resultaat. Binnen Scrum betekent dit dat de sprint een uitdaging moet zijn. Een uitdaging, waarbij je de lat net iets hoger legt dan je denkt gemakkelijk te kunnen halen. Louis van Gaal gaf hiervan een mooi voorbeeld. Hij vroeg tijdens een lezing een persoon zich maximaal uit te rekken. Toen vroeg hij: 'Kun je nog hoger?'. En jawel, hij reikte nog wat hoger. Toen vroeg onze Louis: 'Waarom deed je dit niet meteen?'. Leg dus de lat bewust hoog en ga er vanuit dat je het haalt.

Je focus bij Scrum moet liggen op het moment na de presentatie van de software, als je met elkaar staat te genieten van het feit dat je een goede presentatie hebt gehouden. Waarom? Bij volleybal zijn heel wat teams in de fout gegaan door zich op het behalen van een finaleplaats te concentreren. Toen dat doel bereikt was, →



ging het alsnog mis. Het team verloor de finale. Joop Alberda, coach van het gouden volleybalteam had zijn focus gelegd op de prijsuitreiking na de finale: en zie, het werkte!

D. Evalueer je successen

Een andere belangrijke les die ik van Joop Alberda geleerd is het kiezen van het juiste moment voor evaluaties. Deze dienen bij voorkeur niet te worden gehouden na teleurstellende resultaten. De verwijten zullen dan over de tafel vliegen en de teamleden zullen elkaar niet verder helpen. Wanneer dan wel? Na een succesvolle prestatie. Dan zijn teamleden in 'the winning mood' en staan ze open voor feedback en het doorvoeren van verbeteringen. Dus let op in welke stemming je Scrumteam is als je aan de retrospective begint.

Wat kun je als Agile tester meenemen van een volleybalcoach?

- Een succesvol team heeft aandacht voor het individu. Geef elkaar de ruimte voor inbreng en het geven van feedback, toon interesse in elkaars persoonlijkheid en zorg dat er onderlinge afstemming is. Een gezamenlijke focus op een uitdagend einddoel waarbij je elke stap het beter doet dan de stap ervóór, gaat je verder helpen. Dit geldt niet alleen voor de daily stand-up, maar in alle gevallen waarin je samenwerkt.
- De mate van succes hangt af van het inrichten van een proces dat gedragen wordt door het team. Elkaar letterlijk diep in de ogen kijken helpt en zal de uiteindelijke kwaliteit (en dus ook het resultaat) ten goede komen. Als de afstemming, technisch en persoonlijk klopt, dan is de kans op succes groot.

Laten we ons dan maar aan de wijze woorden van Pythagoras vasthouden: 'Alles wat wij doen is een proces'. Zo kun je op een andere manier kijken naar een volleybalwedstrijd of het functioneren van een Scrum-team. ←

AGILE TESTING ISN'T RISKING IT!

Door Bram Bronneberg • bram.bronneberg@logica.com •  @brambronneberg



Heb je het gevoel dat in jouw Agile project de risico's meegenomen worden in de teststrategie? Ja? En wordt deze strategie dan ook vertaald naar een werkbare aanpak, waarbij de testinspanning verdeeld wordt op basis van de risico's? Als je op één van deze vragen toch nee hebt moeten antwoorden nodig ik je uit om verder te lezen.

De afgelopen jaren ben ik bij tientallen projecten betrokken geweest waarin een Agile component zat. In de meeste gevallen zag ik dat de mannen en vrouwen op de vloer - de programmeurs en testers - volgens Agile principes werkten en de wereld om hun heen dat nog niet helemaal deed. De problemen die hierdoor naar voren kwamen gingen natuurlijk veel verder dan alleen testmanagement en productrisico mitigatie. Ik wil jullie meenemen in die specifieke problemen en de oplossingen die ik daarvoor heb gevonden.

Waar hebben we het over?

We hebben het dus over Agile, Risk & Requirement Based Testen en Product Risico Analyses, wat was dat ook al weer?

Agile

Ik ga hier geen handboek Agile schrijven, maar hier toch even de kaders voor de rest van het verhaal. In de omgevingen waar ik de problemen ben tegengekomen wordt gebruikgemaakt van een Backlog met daarin User Stories. Dit klinkt allemaal heel spannend, maar het is niets anders dan een geordende lijst features welke als losse delen aan de klant opgeleverd kunnen worden.

Risk & Requirement Based Testen

Ook hier alleen de kern van RRBT. Dit is een testmanagementmethode welke een aanpak beschrijft waarin de product gerelateerde risico's centraal staan die de organisatie loopt bij de implementatie van een informatiesysteem. Daarin komen de volgende belangrijke punten naar voren:

- niet alleen focus op de requirements maar ook op de productrisico's;
- testactiviteiten worden op basis van de productrisico's geprioriteerd;
- door te kijken vanuit de productrisico's komen eventueel gemiste requirements boven tafel (en visa versa);
- men spreekt in de taal van de stakeholders, namelijk productrisico's.

Product Risico Analyse

Wat is nu eigenlijk een PRA? Dit is een proces of methode waarmee je de productrisico's van het product analyseert, met als doel de testinspanning met de meest toegevoegde waarde in te zetten. Productrisico's zijn hele andere risico's dan projectrisico's.

Wat is het verschil? Een projectrisico slaat op risico's die de uitvoering van het project zelf in gevaar brengen terwijl een productrisico slaat op de kans dat een product faalt in relatie tot de verwachte schade wanneer het falen in productie optreedt. Uiteindelijk wordt in de PRA bij iedere Requirement een risicoklasse gegeven. Een risicoklasse kan middels 'High', 'Medium' en 'Low' weergegeven worden, maar ook in kwadranten geplott worden. →

Waarom voeren we eigenlijk een dergelijke PRA uit? Zoals in de definitie al gegeven is, willen we de testinspanning op de meest waarde toevoegende manier inzetten. Dit is een manier om verstandig om te gaan met de beschikbare tijd, scope, geld en kwaliteit.

Tegen welke problemen liep ik aan?

Okay, er zijn dus al theorieën en methoden die toegepast worden. In de praktijk worden ze echter niet altijd even effectief toegepast. Dit omdat maar een deel van de theorie gebruikt wordt of omdat er verkeerde interpretaties gemaakt worden. De volgende problemen kwam ik dan ook regelmatig tegen.

Splitsen of samenvoegen van features verstoort eerste inschatting

Ik zag dat er prachtige PRA's gedaan werden op een hele lijst van features, maar later bleek dat het uiteindelijke Scrum-team deze lijst van features niet één op één kon gebruiken als Backlog. Het Scrum-team had namelijk 'User Stories' nodig, welke als een afgebakend stukje functionaliteit in twee tot vier weken opgeleverd kon worden. Dit betekende dat een aantal features samengevoegd of opgesplitst moest worden, waardoor de Risicoklassen die eraan verbonden waren niet altijd doorvertaald konden worden naar de User Stories.

De PRA wordt niet iedere iteratie bijgewerkt

In de meeste PRA methoden benoemt men de noodzaak van het regelmatig bijwerken van de PRA. In traditionele omgevingen is dit principe echter regelmatig verwaterd en zal een PRA maar al te zelden bijgewerkt worden. Omdat in Agile omgevingen de klant zoveel kan sturen, ook op scope, bestaat het Risico dat de PRA al snel niet meer accuraat is. Ben je dus bewust van de inzet die het iteratief bijwerken vraagt van de testmanager maar vooral van de stakeholders om met de hartslag van het Scrumteam mee te bewegen.

De PRA wordt met oogkleppen op gedaan

In de meeste PRA methoden kijkt men vanuit de requirements naar mogelijke productrisico's. De valkuil hierbij is dat men met een soort oogkleppen naar de productrisico's gaat kijken. In sommige methoden zoals RRBT wordt hier rekening mee gehouden en ontdekt men gemiste requirements. In een Agile omgeving is het nog lastiger om niet in deze valkuil te trappen. Hierbij is de focus namelijk continu op de backlog van requirements en zo worden productrisico's regelmatig over het hoofd gezien.

De uitkomst van de PRA wordt niet meegenomen in de planning meeting

Ik zag ook dat bij de planning meetings geen rekening werd gehouden met de aanpak die bij een Risicoklasse hoort. Dit had als gevolg dat een sprint belading qua testwerk regelmatig onderschat werd en een sprint dus ook niet 'gehaald' werd.

Testactiviteiten worden geprioriteerd op basis van productrisico's, onafhankelijk van de sprint backlog

In een poging om de Risico's toch mee te nemen in de Agile werkwijze heeft men geprobeerd om niet alleen de requirements maar ook om de productrisico's te prioriteren. Dit hield in dat men met behulp van bijvoorbeeld MoSCoW probeerde de Must productrisico's als eerste af te dekken. Hierin schuilt echter een onmogelijkheid. Het is namelijk niet zo dat een Must productrisico ook gekoppeld is aan een Must Requirement. Het is dan dus ook goed mogelijk dat een Must productrisico helemaal niet getest hoeft te worden omdat deze alleen verbonden was aan een Would Requirement die uiteindelijk helemaal niet gerealiseerd is. →

Hoe kan het dan wel?

We hebben de belangrijkste problemen zojuist besproken. Nu wil ik jullie graag laten zien hoe het wel kan, gebaseerd op mijn ervaringen.

Fase 1 - Verzamelen Risico Items

De eerste fase van de PRA bestaat uit het verzamelen van de Risico items. Onder Risico items verstaan we de requirements én de productrisico's. Klinkt misschien een beetje typisch dat we ook requirements Risico items noemen, maar dat is niet voor niets. Uit RRBT leren we namelijk dat een productrisico niet bestaat zonder requirement en vice versa.

De eerste stap is het verzamelen en ordenen van de requirements. Deze verzamelen we normaliter vanuit een vorm van documentatie. Als we deze ook nog eens getoetst hebben, kunnen we ze gaan ordenen. Traditioneel spreken we niet over ordenen maar over prioriteren met methoden als MoSCoW. Binnen Agile gebruiken we echter een Backlog welke geordend wordt. Wat we als eerste gaan realiseren is namelijk niet alleen afhankelijk van de business prioriteit, maar we nemen ook zaken mee als afhankelijkheden met andere ontwikkeltrajecten. En tevens zoiets basaal als een makkelijke Requirement om er als team een beetje in te komen. Het is essentieel dat de requirements die we hier verzameld hebben, ook uiteindelijk in alle testsoorten gebruikt worden om te voorkomen dat de PRA op de werkvloer niet meer bruikbaar is. Je spreekt dus niet meer over losse business, user, solution en system requirements - maar over requirements.

De tweede stap is het verzamelen van de productrisico's. Deze verzamelen we normaliter middels interviews en workshops met de verschillende stakeholders. Het is belangrijk dat iedereen zijn of haar zorgen uitspreekt. De business maar ook beheerders en ga zo maar door. Het is aan de testmanager om dit proces te begeleiden en de goede vragen te stellen, zodat ook echt het onderste uit de kan gehaald wordt.

Fase 2 - Calculeer Risico Items

De tweede fase van de PRA bestaat uit het calculeren van de verzamelde Risico items uit de vorige stap. Deze hele fase doe je samen met de stakeholders.

De eerste stap is het bepalen van impact en kans factoren om het waarderen van de Risico's gestructureerd te kunnen ondersteunen. We zien Risico's namelijk als een samenspel tussen de kans dat een product faalt en de te verwachte schade wanneer het falen in productie optreedt. En omdat het lastig is om de vragen: 'Wat is de verwachte schade wanneer...' en 'Wat is de kans dat...' te beantwoorden, breken we deze op in factoren die we wel als losse vragen kunnen beantwoorden. Enkele voorbeelden van impact factoren zijn: Gebruik, Zichtbaarheid en Faalkosten. Dan zou één van de vragen kunnen zijn 'Hoeveel gebruikers zijn er voor dit onderdeel?' Maar wat is dan het antwoord? Om het calculeren later mogelijk te maken, sturen we de antwoorden door deze als meerkeuze op te stellen. Bij Gebruik zou dit dan '100 of meer', 'tussen de 10 en 100' en 'minder dan 10' kunnen zijn. Enkele voorbeelden van kansfactoren zijn: Complexiteit, Hergebruik, Koppelingen, Afhankelijkheden en Grootte. Dan zou één van de vragen kunnen zijn: 'Hoeveel koppelingen heeft dit onderdeel met andere systemen?'. De antwoorden zouden: 'meer dan 5', 'minder dan 5' en 'Geen' kunnen zijn.

De tweede stap is het wegen & calculeren van de Risico items en factoren. In de tabel zien we in het midden een kolom requirements. Aan de linkerkant worden de kans factoren, aan de rechterkant de impactfactoren gewaardeerd. De tabel blijft heel abstract, maar geeft weer hoe de waarderingen per factor tot een totaal komt voor respectievelijk kans en impact, welke samen de Risicoklasse van het Requirement maakt. →

KANS				REQUIREMENT	RISICO KLASSE	IMPACT			
Factor A	Factor B	Factor C	TOTAL			TOTAL	Factor A	Factor B	Factor C
High	High	High	High	REQ 01	Medium	Low	Low	Low	Low
Low	Medium	Medium	Low	REQ 02	Medium	Medium	High	High	Low
Medium	Medium	Low	Medium	REQ 03	Medium	Medium	Medium	Low	Low
High	High	Medium	High	REQ 04	High	Medium	High	Medium	High
High	Low	Low	Medium	REQ 05	Medium	Medium	Low	Medium	Low
Medium	Low	Low	Low	REQ 06	Low	Low	Low	Low	Medium
Medium	Medium	Medium	Medium	REQ 07	High	High	Medium	High	High
Low	Medium	High	Low	REQ 08	Low	Low	Medium	Low	Low
Low	Low	Low	Low	REQ 09	Low	Low	Medium	Low	Low
Medium	Medium	Medium	Medium	REQ 10	Medium	Medium	Low	Medium	Medium

In de projecten waar ik deze methode heb ingezet ben ik gestart met een eenvoudige variant. Daarin heb ik middels een spreadsheet programma voor kans en impact de factoren opgeteld en gedeeld door het aantal factoren. Daarbij heb ik 'Low', 'Medium' en 'High' omgezet naar 1, 2 en 3. Per project werden wel net weer andere factoren gebruikt, afhankelijk van de situatie. Je kunt je echter voorstellen dat men niet iedere factor even zwaar mee wil nemen. Dit zal je met de stakeholders moeten afstemmen. Na deze fase weten we per requirement wat de Risicoklasse is, die we later kunnen gebruiken om de aanpak op de werkvloer te sturen.

Fase 3 - Plotten van Risico Items

De derde fase van de PRA is het plotten van de productrisico's op de requirements en vice versa. Hier laten we heel even de Risicoklasse los en gaan we net dat stapje verder om alle productrisico's af te kunnen dekken. Het plotten gaat als volgt in zijn werk. Je begint met de lijst van requirements en legt de lijst met productrisico's er naast. Nu ga je per requirement de hele lijst met productrisico's door en je noteert alle matches. Als je de hele lijst hebt gehad, ga je het ook nog eens andersom doen. Dus ga per productrisico de hele lijst met requirements door en je noteert weer alle matches. Daarnaast stel je jezelf bij ieder productrisico de vraag of je hiervoor wel een dekkende set requirements voor hebt staan. Je zult zien dat je bij sommige productrisico's geen match kan vinden, dan moeten alle alarmbellen gaan rinkelen, want dan mis je een requirement. Uiteindelijk ben je klaar met het plotten en hou je een matrix over zoals hier weergegeven. →

Product-risico Req- uirement	PR 01	PR 02	PR 03	PR 04	PR 05	PR 06	PR 07	PR 08	PR 09
REQ 01	•						•	•	•
REQ 02	•					•	•		
REQ 03		•							
REQ 04		•	•						
REQ 05				•					•
REQ 06					•		•		
REQ 07	•					•			
REQ 08				•					•
REQ 09			•					•	
REQ 10	•	•	•						

Fase 4 - Van Strategie naar Tactiek

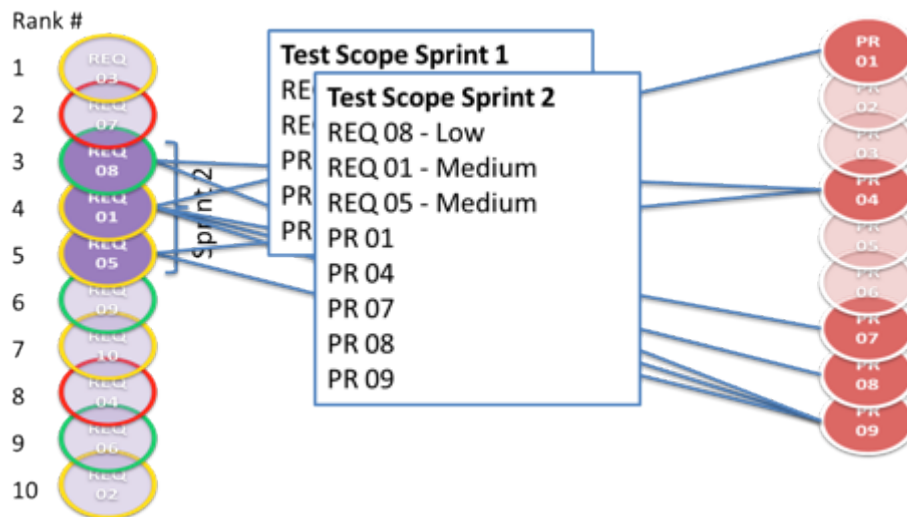
Nu we weten wat de productrisico's zijn en hoe deze zich verhouden tot de requirements, kunnen we spreken van een strategie. Aan een strategie alleen hebben we niets, deze moeten we vertalen naar een goede werkbare tactiek zodat de teams ook daadwerkelijk de principes van de strategie kunnen uitvoeren. Eén van de belangrijkste onderdelen gaat over communicatie: het zorgen dat iedereen hetzelfde doel voor ogen heeft en houdt. De meeste Agile teams werken met een dagstart bord. Een plek waar iedere dag besproken wordt wat men gisteren gedaan heeft, vandaag gaat doen en welke issues men heeft. Dit is dus dé plek waar ook onze tactiek gecommuniceerd wordt.

Zoals we in het begin al gezien hadden, draait het hele idee er om dat we een gedifferentieerde aanpak willen hanteren. Dit doen we op basis van de risicoklasse. We hebben drie klassen onderkend, namelijk 'High', 'Medium' en 'Low'. Bij ieder van deze klassen koppelen we andere acties en deze plakken we in een grote tabel op het dagstartbord. In deze tabel zie je enkele activiteiten die per risicoklasse een andere diepgang voorschrijven.

	High	Medium	Low
Testbasis toetsen middels	Inspectie	Walkthrough	Peer review
UT testgeval review	Testlead	Tester	-
ST testgeval Review	Testlead	Tester	-
Testgeval opnemen in regressieset	Sprint + release	Release	-
Testgeval automatiseren?	CodedUI	Record & Playback	-
Testuitvoer op browsers	Top 3	Top 2	Top 1



Het zal je opgevallen zijn dat er geen diepgang per testtechniek in staat. Dit is wel een traditionele manier om de testinspanning te verdelen, maar in mijn ogen doe je daar testers tekort mee. Een tester is een vakman en is dan ook zelf in staat de juiste testtechniek te kiezen en toe te passen. Je schrijft een timmerman ook niet voor om een klauwhamer of een moker te gebruiken. Naast de activiteiten tabel hebben we ook nog een extra informatiebron om het team de juiste tests te laten analyseren en uitvoeren. We weten namelijk per sprint welke requirements we gaan oppakken. Met behulp van onze matrix weten we dan ook welke productrisico's we moeten afdekken. In dit diagram wordt dit duidelijk weergegeven. Plak ook deze uitsnede van de risicomatrix op het dagstartbord en iedereen ochtend wordt iedereen eraan herinnerd.



Nu het dagstartbord vol staat met verwijzingen naar de productrisico's en iedereen op de hoogte is van de acties per risicoklasse, kunnen we spreken over een gedifferentieerde testaanpak.

Dus in een notendop

Nu je het hele verhaal hebt gelezen, heb je een beeld hoe ik deze methode toepas in mijn projecten. Laten we nog kort maar krachtig bij alle onderdelen stil staan.

Fase 1 Verzamel Risico Items

- Verzamel & Order Requirements
- Verzamel Productrisico's

Fase 2 Calculeer Risico Items

- Bepaal Impact & Kans Factoren
- Weeg & Calculeer Risico Item/Factoren

Fase 3 Plot Risico Items

- Plot Requirements & Productrisico's

Fase 4 Van strategie naar tactiek

- Risicoklasse tactiek
- Uitsnede risicomatrix ←

DE ROLLEN VAN DE AGILE TESTER

Door Eric Jimmink • eric.jimmink@ordina.nl en Anko Tijman • Anko.Tijman@ordina.nl



Testen is echt een teamsport geworden: alle teamleden voeren testactiviteiten uit en ook de klant heeft een belangrijke rol bij het testen. Het testen moet zo veel mogelijk van het kritieke pad af, vroeg beginnen en zo veel mogelijk parallel met de bouw worden uitgevoerd. Daarmee is de tijd van validatie door een onafhankelijk QA team definitief voorbij. Veel checks worden nu al uitgevoerd door de bouwers in een multidisciplinair team.

Wat betekent dit voor de tester zelf? Is hij overbodig? Of treedt hij vooral op als testcoördinator? Wij vinden van niet. De tester moet vooral waarde toevoegen aan het team vanuit zijn vak. Je bent meer testspecialist dan tester. Als testspecialist binnen een Agile team heb je verschillende rollen, waar die van testuitvoerder slechts één aspect is.

Wij onderscheiden vijf rollen, waarbij de invulling ervan onder andere bestaat uit:

- Teamspeler: vooral snel kunnen schakelen en feedback leveren waar dat op dat moment nodig is voor het team.
- Analist: de traditionele aspecten, maar ook: in het met elkaar uitwerken van de details van requirements, kom je zeker op het punt dat je samen ontwerpbeslissingen moet nemen.
- Kennisdeler: door testkennis te delen met andere teamleden, stel je hen in staat effectiever te testen.
- Bruggenbouwer: is verbindingen leggen: je communiceert vanuit je testrol met vrijwel alle partijen in de business en het IT project.
- Dirigent: je zorgt ervoor dat je het overzicht bewaart over de testactiviteiten en dat de overige teamleden de juiste dingen tijdig testen.

Ieder van deze rollen zullen we nader behandelen.

De tester als teamspeler

De primaire toegevoegde waarde van het testen in een Agile project is de feedback die gegenereerd wordt door het daadwerkelijk uitvoeren van tests. Daarom is er het Agile paradigma 'test early, test often'. Iedereen binnen het team is gebaat bij snelle feedback: een ontwikkelaar kan een fout veel sneller oplossen wanneer de materie nog vers in het geheugen ligt. Daarnaast wordt door veelvuldig testen voorkomen dat er code wordt gebouwd op code die niet getest is. Veel en vaak testen voorkomt dus gevolgfouten. Last but not least: wanneer je de tests veelvuldig uitvoert op de testomgeving, dwingt dit het team tot het frequent integreren van de code ('continuous integration'). Dit zorgt ervoor dat de verschillen in code tussen de ontwikkelaars minimaal zijn en dus dat conflicterende issues snel gevonden worden. Daarmee blijft de kwaliteit van de code dus zo stabiel mogelijk.

Een belangrijke bron van informatie voor de teamspeler is de stand-up meeting. Je hoort bijvoorbeeld welke (sub)delen van functionaliteit testbaar zijn voor jou als tester. Dan kun je dus vroege feedback genereren door de automatische of handmatige tests uit te voeren. →



Wanneer je dat doet, kun je ook vaak extra testgevallen opstellen die de diepgang van de test verhogen. Daarnaast komen knelpunten ook aan de orde in de stand-up meeting en de tester kan dan aanbieden om samen het probleem te bekijken. Uiteraard alleen wanneer jij denkt dat je toegevoegde waarde kunt leveren. Gelukkig leert de praktijk dat de denkwijze van de tester hierin veel waarde toevoegt.

Van de tester wordt dus verwacht dat je inspeelt op wat er in het team gebeurt. Daarmee hebben we een natuurlijk spanningsveld met de aard van het testen: je wilt het liefst in alle rust testen op een stabiel systeem in plaats van in de hectiek van een project dat werkt aan 'groeierende' software. Daarom is het goed om met jouw team duidelijke werkafspraken te maken: soms heb je als tester rust en stabiliteit nodig en vaak moet je snel kunnen reageren op veranderingen.

Naast de tester die als teamspeler gericht is op het team, gaat het bij je rol als teamspeler ook om de andere kant op samen te werken. Je zult soms testtaken kunnen, willen of moeten overdragen aan een teamlid die geen tester is. Een veel voorkomende aanleiding van het delegeren van testtaken is de tijdsdruk die vaak optreedt wanneer het einde van een iteratie nadert. Wanneer de tests eenvoudig en inzichtelijk zijn opgezet, vergemakkelijkt dit het delegeren naar een collega die zelf geen testspecialist is. Eenvoud in de beschrijving van de tests is dus een doel op zich, omdat het de communicatie bevordert. Bij het overdragen van testtaken aan niet-testers zijn de volgende aspecten belangrijk:

- het is duidelijk wat de noodzakelijke input is en wat er aan output verwacht wordt;
- de taak is zodanig opgezet dat het ook makkelijk over te dragen en uit te voeren is;
- het teamlid heeft de taak al eerder gedaan, bij voorkeur samen met jou als tester;
- je hebt bij de voorbereiding van de tests andere teamleden betrokken.

Op deze manier kun je er zeker van zijn dat de rol als tester goed wordt uitgevoerd.

Zoek de activiteit met de op dat moment hoogste toegevoegde waarde

Een algemeen Agile principe is dat ieder teamlid voortdurend die actie moet uitvoeren die op dat moment de meeste waarde toevoegt voor het team. Doorgaans zal dat het oppakken van een taak zijn, horende bij de belangrijkste user story die nog niet is afgerond. Soms zul je een taak oppakken buiten je eigen specialisme. Zo kan het gebeuren dat je als tester niet alleen tests uitvoert en helpt om requirements beter te maken, maar dat je bijvoorbeeld ook code gaat reviewen, documentatie schrijft of zelfs fouten oplost.

Eric: Tegen het einde van de iteratie zag ik dat het team geen voortgang wist te maken met het oplossen van een ernstige fout die ik een week tevoren had gemeld. Ik stelde mijn testwerk uit en vroeg of ik misschien iets kon doen om te helpen met het zoeken naar een oplossing. De ontwikkelaar liet me op zijn machine zien wat er mis ging. Dat gaf me een idee, en na even zoeken op Internet vond ik de tip die leidde tot het verhelpen van het probleem.

Anko: Ik gebruik het Scrumboard als heilige graal voor mijn prioriteiten. Daarop is vaak heel goed te zien waar knelpunten zitten. Zo staan sommige taken soms lang open en kun je als tester meedenken met de oplossing of achter een stakeholder aangaan. Ook zijn opstoppingen in het proces goed te zien wanneer er teveel taken op een bepaalde status blijven staan. In een Agile team moet er een flow zijn en daarin speel jij als tester ook een rol.



De tester als analist

In een Agile project werk je multidisciplinair samen. Dit geldt dus ook voor de requirements: als tester denk je mee met de gebruikers, de analist/ontwerper en de ontwikkelaars. De praktijk leert dat het buitengewoon moeilijk is om specificaties in één keer correct, consistent en volledig op te stellen. Toch is het nog veelal gemeengoed om een document als solist uit te werken. Vanuit het oogpunt van volledigheid en juistheid van specificaties is het veel gemakkelijker en efficiënter als de auteur van de specificaties in een vroeg stadium om feedback vraagt van andere teamleden. En dan gaat het met name om de feedback vanuit de andere disciplines. Als analist raak je zonder feedback gemakkelijk in een gedachtecronkel en mis je het overzicht op de impact van de beslissingen die je neemt op de overige disciplines. Dit geldt dus niet alleen voor feedback vanuit het testen, maar ook vanuit de ontwikkelaars, de implementatiespecialisten en de beheerders.

Als tester hoef je je in een Agile team dus niet te beperken tot het reviewen van de requirements op volledigheid en consistentie: je mag ook zelf meedenken, suggesties doen en risico's benoemen. Natuurlijk wordt er van je verwacht – net zoals dat ook voor de andere teamleden geldt – dat je het domein je snel eigen maakt (om meerwaarde voor de business te kunnen genereren) of dat je je technisch verdiept (om meerwaarde voor de IT'ers te leveren). De keuze voor welke richting je kiest hangt af van je competenties en interesses.

Als tester in een Agile project zul je veelal werken met requirements die slechts op hoofdlijnen zijn uitgewerkt. Het is zaak om ook die ruwe requirements te reviewen en daar feedback op te geven, nog voordat ze de iteratie in gaan. Deze voorbereidende activiteiten zijn belangrijk: als requirements in de iteratie worden opgepakt die nog niet 'ready' zijn, dan treedt er onherroepelijk vertraging op. Of er wordt iets gebouwd waar de klant geen behoefte aan heeft. Een bepaalde mate van (functionele!) details is dus noodzakelijk. Het uitwerken van de requirements vanuit je rol als tester kan op meerdere manieren:

- het ondersteunen van het SMART maken van de requirements;
- het met het team verder uitwerken van een schermontwerp;
- het toepassen van het 'Story tests' principe (zie Crispin, 2009).

Requirements moeten in elk geval duidelijk genoeg zijn voor de beoogde lezers (het team, met name), en geen ruimte bieden voor verschillende interpretaties. Het is belangrijk om te voorkomen dat er in de verwoording van de wensen van de business aannames staan over de technische invulling. Specificaties zonder een voorgesorteerde oplossingsrichting geven het team meer ruimte voor een betere oplossing. Tom Gilb somt die algemene reviewcriteria op als in de figuur rechts.



Figuur: Criteria Tom Gilb

Het uitwerken van het schermontwerp blijkt in de Agile praktijk een bijzonder leerzame ervaring om concreet toe te werken naar een werkend product. Dit gebeurt uiteraard samen met de product owner en eindgebruiker. Door dit al in een vroeg stadium te doen ontstaat er aan beide zijden (de klant en het ICT-team) een goed beeld van hoe het product er uit moet gaan zien. Dit dient dan als gids voor de gewenste functionaliteit. Daarnaast kun je op die manier al aandacht aan de usability van het systeem besteden, zeker wanneer je de ontwerpen 'klikbaar' maakt. →

Een team kan besluiten om nog een stap verder te gaan door de acceptatietests te concretiseren voorafgaande aan de implementatie. Door stelselmatig de requirements uit te breiden met voorbeelden van het gewenste gedrag van het systeem, en deze voorbeelden te gieten in een vorm waarin ze kunnen worden uitgevoerd als tests, wordt sturing gegeven aan de gehele ontwikkeling. Zo wordt kwaliteit vooraf geborgd en is acceptatie nog slechts een formaliteit. Dit wordt ook wel 'Acceptance Test Driven Development' of 'story tests' genoemd. In de ideale situatie kun je spreken van uitvoerbare requirements – requirements die de vorm hebben van een test, en die eenvoudig genoeg zijn opgezet dat analisten ze kunnen schrijven en mensen van de business ze kunnen lezen. Op die manier houd je de feedback loop kort, en neem je een bron van interpretatiefouten weg. Het team kan zo toewerken naar een oplossing die in één keer wordt geaccepteerd.

Het aanvullen van requirements

Eric: Als tester in een Agile team werk ik regelmatig user stories uit. Zo had ik bijvoorbeeld een aantal data-conversies binnen een project. Onze product owner kon op voorhand niet goed aangeven welke invulling 'goed genoeg' zou zijn. Als team kozen we ervoor om de implementatie pas drie weken later op te pakken, zodat in de tussentijd de details konden worden ingevuld. Toen heb ik de brondata geanalyseerd en relevante documentatie opgesteld. Natuurlijk kwam daar feedback op en is de formulering meermalen aangepast. Tijdens het opstellen van de documenten kreeg ik als vanzelf ook een veel beter beeld hoe ik het testen moest aanvangen. Ik kon doelgericht testen en beter gefundeerde bevindingen rapporteren. Het afwachten tot iemand van de business tijd zou vrijmaken voor het opstellen dan wel uitwerken van de specificatie had ongetwijfeld geleid tot vertraging. Proactiviteit loont.

Anko: Een user story wordt vaak in het volgende format opgezet: 'ALS klant WIL IK een boodschappenmandje ZODAT ik meerdere bestellingen kan doen die ik in één keer kan afrekenen'. Sommige Agile teams werken dit niet verder uit. Maar als tester heb ik dan nog heel veel vragen over de gewenste functionele implementatie, zoals hoeveel producten en tot welke bedrag je maximaal mag bestellen, kun je uit verschillende categorieën kiezen, waar wordt dat winkelmandje getoond, kun je er nog uit verwijderen, etcetera. Dit soort uitwerkingen en beperkingen kun je uitstekend in een aantal testgevallen bij de user story uitwerken. Doe dit samen met het team en de klant. Dit is voor iedereen erg verhelderend!

De tester als kennisdeler

Naast een rol als analist en teamspeler is de tester ook iemand die kennis over testen met zijn teamleden deelt. Het samen testen met een ontwikkelaar is natuurlijk gedrag voor een Agile tester en op die manier leren beiden meer over elkaars vakgebied. Daarmee verhoog je de kwaliteit van het werk van beiden. Als tester kun je een ontwikkelaar meer leren over toe te passen testtechnieken, bijvoorbeeld op unittests. Denk aan beslissingstabellen, pseudo code of grenswaardenanalyse. Of hoe je een goede handmatige blackbox test uitvoert, zoals een Exploratory test charter. Om testtechnieken toegankelijk te maken, hebben we er een aantal samengevat en waar nodig vereenvoudigd zodat ze op één A4-tje kunnen worden nagelezen. Deze zogeheten Quick Reference Cards zijn beschikbaar gesteld op de downloadpagina van onze blog www.Agileopmaat.nl.

Op een hoger niveau wordt kennis van de business en het bedrijfsproces belangrijker; testsituaties worden dan omschreven in de termen van de business en geformuleerd in samenspraak met materiedeskundigen. →

Paarsgewijs testen met een materiedeskundige is dan een logische aanvulling op jouw én op zijn kennis. Als tester heb je een schat aan ervaring in het vinden van fouten in software. Vaak is een materiedeskundige of gebruiker er niet op gespist en valideert men alleen wat wel werkt. Of men denkt voornamelijk vanuit de eigen ervaring en kennis ('error guessing'). Als tester heb je een bredere blik vanuit de techniek en de kennis die je in het ICT-project hebt opgedaan. Daarnaast kun je je testgevallen ter review aanbieden aan de deskundigen, zodat met de feedback die daaruit komt geborgd wordt dat de juiste testen worden uitgevoerd. Zodoende leer je als tester meer over de business en kweek je bij de ander meer inzicht en begrip voor het testen.

De tester als bruggenbouwer

De vierde 'pet' die de Agile tester draagt is het leggen van verbindingen in de organisatie. In een Agile project zal een tester vanuit zijn rol met heel veel partijen willen en moeten communiceren. Het is van groot belang om tijdig met de verschillende stakeholders te hebben gesproken, om een goed beeld te hebben van de risico's en de verwachtingen die er bij hen leven. Dit is namelijk direct van invloed op de teststrategie waarin je bepaalt welke systeemdelen je met welke diepgang gaat testen. Hoe hoger of belangrijker het risico, des te dieper moet de test in verhouding zijn en hoe vroeger de test dient te starten. Daarnaast helpt het nauwe contact om betere sturing op de acceptatietest te geven: je kunt beter vooraf voorspellen wat de klant aan gewenste functionaliteit wil hebben en je kunt het belang van de acceptatietest nog eens benadrukken. Dit is soms noodzakelijk omdat nog niet veel organisaties gewend zijn aan een min of meer continue acceptatietest die per iteratie plaatsvindt. Het voordeel van de continue acceptatietest in een Agile omgeving is echter juist dat er vroegtijdig getest wordt door de klant. Die feedback heb je nodig om je project in de juiste richting te kunnen sturen. Wanneer de klant dus zijn rol als acceptant geen concrete handvatten weet te geven, dan ontbreekt er een stroom aan informatie in het project, die juist heel veel toevoegt aan het project. Deze actieve rol geldt ook voor de toekomstig beherende partij; ook met hen zal de tester een nauwe werkrelatie moeten onderhouden om te voorkomen dat er bij het naar productie gaan nog onverwachts nieuwe eisen of wensen naar boven komen.

Wij zien veel Agile teams en lang niet alle teams hebben de beschikking over een 'eigen' tester. Sommige testers zijn dan ook over meerdere Scrum-teams verdeeld. Hoewel dit zeker niet ideaal is (het schakelen tussen projecten en teams kost naar verhouding veel tijd), heeft het als voordeel dat je een brug kunt slaan tussen de verschillende Scrum-teams. Sommige teams hebben uitstekende unittest suites, terwijl anderen ermee worstelen. Ook in dat soort situaties kun je als tester je toegevoegde waarde leveren door de best practices met elkaar uit te wisselen. Bijvoorbeeld door een kennissessie te organiseren of gewoon een uurtje te plannen tussen ontwikkelaars uit de verschillende teams.

De tester als dirigent

Het testen in een Agile team wordt uitgevoerd door specialisten uit verschillende vakgebieden. De testspecialist behoudt het overzicht over het geheel en helpt anderen een zo goed mogelijke job te leveren. Omdat het testen zo verspreid plaatsvindt en dynamisch van karakter is, is het belangrijk iemand te hebben die de testactiviteiten 'orkestreert'. Zeker in complexere trajecten, waarbij er sprake is van elkaar overlappende systeemtests, non-functionele tests, ketentests en acceptatietest is overzicht onontbeerlijk. De juiste mensen moeten op het juiste moment de juiste dingen doen. Daarom ben je dus ook als het ware een dirigent. →

In beginnende Agile teams zie je vaak dat er slechts één tester is die het werk van een groot aantal ontwikkelaars mag proberen te controleren. Dat moet andersom. De testspecialist bedient de teamleden door met hen te overleggen welke testsituaties zij zelf (geautomatiseerd) kunnen gaan testen. Slechts een klein deel van het totale testwerk zal echt de expertise van de specialist vereisen; daar heeft hij of zij dan wel tijd voor doordat deze het leeuwendeel van de tests heeft belegd binnen het team.

Deze vijf verschillende petten van een Agile tester hebben een belangrijk ding gemeen: het gaat uitdrukkelijk om communicatieve vaardigheden. Dit komt dus bovenop de vaardigheden die geassocieerd worden met het reguliere testwerk. Daarmee kruipt de tester uit zijn schulp: niet alleen draait hij volledig mee als tester in het team, hij gebruikt zijn expertise ook nog eens om anderen hun werk nog beter te laten doen. Dit allemaal om een nog beter, voorspelbaarder en werkbaarder resultaat voor de klant neer te zetten. Daar doen we het immers allemaal voor. ←

HET SPROOKJE VAN DE GOEDE AANPAK EN DE (SL)ECHTE WERELD

Door Egbert Bouman • egbertbouman@valori.nl



Oud, nieuw en slim

Er was eens een Slimme Tester. Hij hield zowel van oude dingen als van nieuwe dingen. Hij had er veel plezier in om oud testen en nieuw testen te combineren. Samen met bouwers en gebruikers. Hij noemde dat 'slim testen'. Dat ging best goed en hij werd vaak om raad gevraagd door vrienden en bekenden die graag oude wijsheden met nieuwe inzichten wilden combineren.

Belangrijk Project

Op een dag werd hij gevraagd door Degelijke Onderneming om Belangrijk Project te helpen met slim testen. Natuurlijk zei hij geen nee want hij bouwde liever echte kastelen dan luchtkastelen. Hij begon maar eens met een aantal belangrijke mensen van het belangrijke project te bezoeken. Ook de Succesvolle Leverancier.

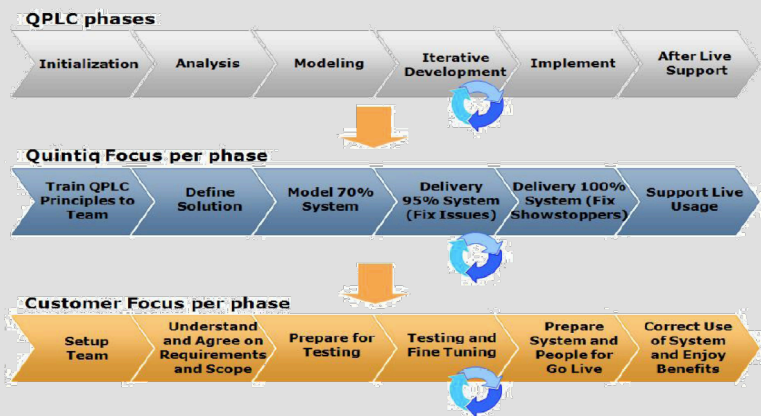
SLAP

Hij was prettig verrast: de Succesvolle Leverancier houdt net zoveel van oud en nieuw als hij. De Succesvolle Leveranciers AanPak (SLAP) is werkelijk een intelligente combinatie van oude en nieuwe principes voor systeemontwikkeling en slim testen. 'Gefeliciteerd, jullie hebben het begrepen' zegt de Slimme Tester en huppelend begeeft hij zich naar de Degelijke Onderneming om zijn enthousiasme voor de SLAP aanpak te delen.

Rechte Paden

Maar dat viel tegen! 'Wij testen hier alleen via de oude, rechte paden' legt de Degelijke Testbaas van de Degelijke Onderneming hem uit. 'Ik heb hier een map, onze Test-Map, en daar staat het allemaal in. →

De SLAP aanpak: een sterke integratie van oud en nieuw



Voor de SLAP aanpak heeft de – inderdaad goede en succesvolle – Quintiq Project Life Cycle aanpak model gestaan. SLAP gaat over configuratie en implementatie van een standaardpakket, zoals de meeste ICT projecten anno nu. De afbeelding laat onder andere zien dat :

- Er goed wordt nagedacht en gemodelleerd vooraf. Een oud en gezond principe!
- Niet wordt getracht in één keer waternval/big bang op te leveren, maar een 70% release. Perfect, big bang lukte vroeger ook al niet, al deden wel we net alsof.
- Iteratief/incrementeel wordt gebouwd en getest na oplevering van de 70% release. Een gezonde Agile benadering.
- Er geen 100% versie naar productie gaat, maar een 95% versie. Terecht, want hoe goed je het ook doet, er zijn altijd dingen die je in de testomgeving niet of niet tegen redelijke kosten kunt zien.
- Teambuilding en 'customer collaboration' veel aandacht krijgen. Hiermee wordt recht gedaan aan een goed en belangrijk principe van het 'Agile Manifesto'.

Wij zijn een Degelijke Onderneming en ook Succesvolle Leveranciers moeten op onze manier werken, anders gaan ze maar ergens anders succesvol zitten wezen'. Bedroefd ging de Slimme Tester in een stiltekamertje zitten. De inhoud van de map was best OK, maar hoe moest hij hiermee oud en nieuw bij elkaar brengen?



Prins Alexander op het witte paard

Buiten liep Prins Alexander op zijn witte paard voorbij en plotseling wist hij het: Prins Alexander diende al jaren de Degelijke Onderneming en stond in hoog aanzien om zijn slimheid, energie en vechtlust. Hij was weliswaar geen vijand van de degelijke testaanpak, maar deed het toch liefst op zijn eigen manier en hij zou zeker bereid zijn om de Succesvolle Leverancier te steunen. En inderdaad: toen Prins Alexander samen met enkele gelijkgestemden zijn invloed deed gelden was onze Slimme Tester al snel waar hij wilde zijn: het belangrijke project ging ook wat testen betreft 100% de SLAP aanpak volgen.

Weerstand

Van verschillende kanten ondervond hij nog wel weerstand. Vooral van de groep die zich 'Alle QA daar' noemde. Die vond dat niemand over QA na hoefde te denken omdat dat centraal was belegd bij de Degelijke Testbaas. Ook had hij te maken met een niet geheel onsympathieke maar te fanatieke Taliban (Test Automation leeft en in business acceptatie noodzakelijk) en van een Scrumdamentalistische splintergroep, die echter nog nauwelijks voet aan de grond had bij Degelijke Onderneming.

De start

Maar de Slimme Tester had er zin in: de SLAP-aanpak kon echt gaan vliegen. Hij keek uit naar de eerste belangrijke mijlpaal: de 70% release, dat wil zeggen het volledige resultaat, met nog 30% gezamenlijk detailleringwerk voor de boeg.

Doornen op het pad

Al snel werd duidelijk dat rozen ook doornen hebben: het moest allemaal snel, de planning was strak en de politieke druk hoog. Maar de Succesvolle Leverancier verzekerde dat niets onmogelijk is en dat ze zich grondig gingen oriënteren op de processen van Degelijke Onderneming. Dat kwam de mensen van Degelijke Onderneming goed uit, want zij hadden het al zo druk en ze wisten nog niet helemaal precies wat ze wilden. Maar daarmee ging de Succesvolle Leverancier gelukkig helpen.

Tromgeroffel en gejuich

De 70% release was af. Beter gezegd: nu moest hij maar eens een keer af zijn. Het duurde veel langer dan voorzien. Maar ja, beter nu goed nagedacht dan later alles overdoen. De Slimme Tester zat nog even in zijn stiltekamertje. Hij had hard gewerkt om de eisen voor de 70% demo goed vast te leggen. Dat viel nog niet mee, maar nu was hij redelijk tevreden. Als aan deze eisen werd voldaan, dan kon de expeditie zonder al te veel risico door met de volgende etappe. →

Ain't Talkin', Just Walkin'

De 70% demo verliep rommelig maar voldeed duidelijk niet aan de eisen. En toen gebeurde er iets wat heel vreemd was, maar misschien toch ook weer niet: de modelleerfase werd afgesloten en de iteratieve ontwikkelfase ging van start met een model dat niet 70% was maar hooguit 30%. Er werd niet eens echt over gepraat, de expeditie zwoegde gewoon verder. In de hoop dat in de iteratieve fase meer vaart gemaakt zou kunnen worden, waardoor de achterstand wel weer goedgeemaakt kon worden. De psychologische en politieke druk van de planning woog zwaarder dan de feitelijke voortgang. De Slimme Tester telde zijn knopen: 'goede aanpak + verkeerd gedrag = geen resultaat'. Hiermee was de trend voor Belangrijk Project definitief gezet. 'Dit gaat vast niet meer goed komen' somberde hij. Echt geluisterd werd er niet. Hij wilde zijn vrienden niet in de steek laten en voegde zich weer bij de groep.

Pas op de plaats

Dapper zwoegde de expeditie verder maar het bos werd steeds dichtter. Ze kwamen wel dichtter bij hun doel, maar de afstand was nog ver en navigeren werd steeds moeilijker. De bevoorrading was steeds minder geworden en van SLAP was weinig meer te merken. Er ontstonden kleine conflictjes en de sfeer verslechterde. Enkele maanden later zat een groep mensen van Degelijke Onderneming bij het kampvuur. De mensen van Succesvolle Leverancier hadden elders hun kamp opgeslagen. De stemming was bedrukt: ze hadden wel dingen bereikt maar het grotere beeld was dat de slag van het Belangrijk Project nog lang niet was gewonnen. Laat staan de oorlog voor Degelijke Onderneming.

Terugblik

Ze blikten terug op hoe het allemaal was gegaan. Ze trokken samen conclusies, De aanpak was nog steeds goed en ze stelden vast dat de testaanpak en de gezamenlijke focus ook goed waren geweest. Maar de uitvoering wilde niet lukken. Een schrale troost, want als je niet samen wint verlies je allemaal. Hoe kwam dat nou? →

Een deel van de conclusies

- Basis niet stabiel: de mensen van Degelijke Onderneming hadden nog geen eenduidig beeld van de strategische keuzes op het punt van centralisatie/decentralisatie
- Complexiteit van interfaces onderschat (zoals nog te vaak gebeurt) en onvoldoende in kaart gebracht
- Mijlpalen niet gerespecteerd: politieke druk en deadlines wegen zwaarder dan feitelijke voortgang. De aanpak bood de mogelijkheid om tijdig in te grijpen maar die kans is niet benut
- Te veel overgelaten aan leverancier, die te veel belofde met te weinig detailkennis van Degelijke Onderneming
- Geen stabiele bezetting van het projectteam, van beide kanten.
- SLAP en het oude Degelijke Proces (door allerlei betrokkenen nog gehanteerd) onvoldoende op elkaar afgestemd
- Kwaliteit van de analisten en (business) consultants niet altijd op niveau en business consultants geen goede linking pin: staan dicht bij bouw dan bij business
- Datasets voor validatie en testen niet tijdig beschikbaar. Belang van testdata wel tijdig benoemd, maar hoeveelheid werk toch nog onderschat
- Sleuteldocumenten niet tijdig gereed en niet goed beheerd in samenhang met de applicatie en openstaande issues
- Afspraken wel gemaakt maar onvoldoende vastgelegd voor beoordeling/acceptatie later in het traject
- Onduidelijkheid over precieze omvang en diepgang van deliverables: 'Definition of Done' node gemist (belangrijk Scrum principe)
- Door onvoldoende overzicht en tijd zijn dingen nodeloos ingewikkeld gemaakt en door de ontstane complexiteit werd het moeilijk om hoofd- en bijzaken te onderscheiden en issues te prioriteren
- De SLAP aanpak benadrukt het belang van teambuilding en kennisdeling, maar dat is onvoldoende gedaan
- Leverancier testte zelf onvoldoende, waardoor teveel zware issues in Customer testen

Conclusies

De drie hoofdconclusies waren als volgt:

- De SLAP aanpak is OK;
- Oud of nieuw: de gewone basishygiëne voor projecten is doorslaggevend;
- Niet plannen en intenties zijn bepalend, maar gedrag in de dagelijkse uitvoering.

En vooral: *sprookjes bestaan niet.*

Ze waren ondanks alles van plan hun voordeel te doen met alle waardevolle ervaringen en nog lang en gelukkig te leven.

Noot van de auteur: Zoals dat hoort in sprookjes zijn de personages in dit vergaand versimpelde verhaal grove karikaturen. De Slimme Tester is zelfs een combinatie van meerdere personen. Er bestaat slechts een vage gelijkenis met de uitermate professionele, verstandige en waardevolle mensen die ik in de Echte Wereld tegenkom. Maar de boodschap is serieus, dat wel!

(Met dank aan collega slimme testers Patrick Smulders en Teun van Horssen) ←

LEER LEAN TESTEN, SPEEL DE KANBAN GAME

Door Egbert Bouman • egbertbouman@valori.nl

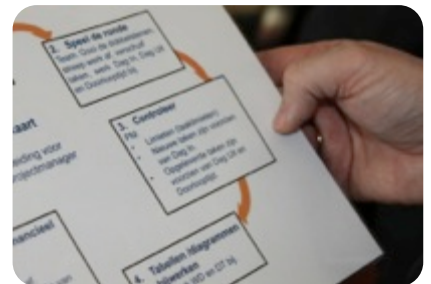


Collaboration is key. Als testers moeten we steeds meer samenwerken in Agile, multidisciplinaire ontwikkelteams. Ik hoop dat we dat ook willen, want testers horen niet in een ivoren toren en 'collaboration is key'. Daarom is het essentieel dat we ons oriënteren op moderne Agile en lean principes en de populaire aanpakken die daarbij horen. Niet om altijd klakkeloos de hippe ideeën van developers over te nemen. Wel om het goede van ouderwets, degelijk testen effectief te combineren met het goede van moderne aanpakken en samenwerkingsmodellen.

Om heel eerlijk te zijn: toen ik een jaar of vier geleden voor het eerst van Kanban hoorde, was mijn eerste gedachte 'OK, Scrum is zeker niet hip genoeg meer en daarom hebben de wereldverbeteraars weer wat nieuws verzonnen'. Zo gaan die dingen. Maar wat bleek? Kanban bestaat al lang. Maar het is de laatste jaren, voor mijn gevoel vrij plotseling, populair geworden vanwege de eenvoudige invoering, sterke visualisatie en de nadruk op samenwerking en het toevoegen van stakeholderswaarde. Inmiddels ben ik om, Kanban is een aanwinst!

Wat is Kanban precies?

Kanban wordt toegepast in combinatie met of als alternatief voor Scrum. Het is een lean aanpak voor software ontwikkelteams, afgeleid van het Toyota Production System en Eliyah Goldratt's Theory of Constraints. Kernpunten van Kanban zijn flow (doorstroming), transparantie en het pull-mechanisme. Het pull-mechanisme werkt niet vanuit de vraag, maar vanuit de maximale flow die het team over een langere periode kan behappen. Daarbij zorgen de zogeheten WIP limieten voor een gebalanceerde vulling van de pipeline. WIP staat hier voor 'Work In Progress'.



Testen is waste?

Een effectief testproces heeft veel baat bij het toepassen van lean principes. Maar Kanban komt primair uit de 'manufacturing' hoek en geforceerd toepassen van lean principes op het testproces kan verkeerd uitpakken. Het kan zelfs gevaarlijk zijn voor ons testers, want testen kún je zien als waste. En als het team 'first time right' oplevert dan moet die waste geëlimineerd worden. Okay, als ...



Continue verbeteren

Toch kunnen testers hun voordeel doen met Lean, vooral omdat het primair gaat over continue verbeteren. Testers en ontwikkelaar kijken elke dag opnieuw hoe ze samen maximale voortgang kunnen boeken. De ervaring in organisaties, ook grote organisaties zoals ASR heeft uitgewezen dat Kanban zeer geschikt is voor releasematig werken en beheer. →

Kanban is een kijkspel

Kanban is in hoge mate visueel. Sterker nog Kanban is Japans voor 'Visueel bord' of 'Visuele kaart'. Daarom is het spelen van een spel bij uitstek de manier om Kanban te leren kennen. De implementatie in de echte wereld staat namelijk heel dicht bij het spel. Om zicht te krijgen op de flow maakt Kanban onder meer gebruik van een whiteboard met kolommen en swimming lanes. Elke taak wordt gevisualiseerd met een kaartje (of post-it) die zich altijd in één van de kolommen op het bord bevindt: backlog, design, development, test, en deployed.

Strijdende teams

In de Kanban Game wordt een praktijksituatie gesimuleerd, waarbij sprake is van een IT-waardeketen met vier verschillende rollen. De strijdende teams moeten niet alleen het spel zo snel mogelijk afmaken, maar daarbij ook proberen om zo veel mogelijk waarde te creëren voor 'hun' klant. Het hele team moet door intervisie bepalen hoe de werkpakketten zo snel mogelijk te realiseren gegeven een 'Work In Progress' limiet. Dit vereist strategisch denkwerk en intensieve samenwerking om zo de juiste prioriteit te stellen en de hoogst mogelijke productiviteit te realiseren. Dat levert een leuke strijd op. Een foto-impressie van het spel is te zien op www.valori.nl/mom.

Teambuilding plus

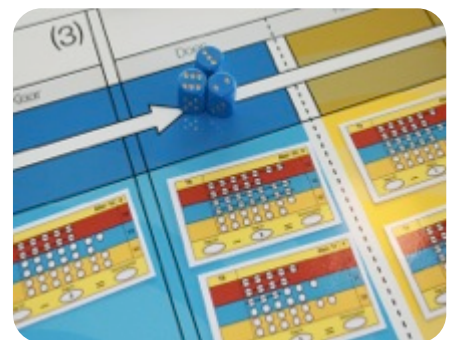
Het spel onderscheidt zich van gebruikelijke teambuilding-sessies en -games doordat gestuurd wordt op flow. Ieder mens is het meest productief als hij voor maximaal twee opdrachten tegelijkertijd verantwoordelijk is. Deze limiet creëert een mindset waardoor mensen veel meer gaan nadenken over h^oe zij van waarde kunnen zijn voor de opdrachtgever en gebruikers. De deelnemers krijgen daar enorm veel energie door. Je ziet vanzelf verbetervoorstellen ontstaan. Deze game is dan ook een goed hulpmiddel voor organisaties die Lean (meer) in hun IT-waardeketen willen gaan toepassen.'



Ontdek Kanban op het TestNet najaarsevent

Het door Valori ontwikkelde Kanban bordspel is inmiddels door honderden deelnemers gespeeld en die zijn laaiend enthousiast. Wij ontwikkelden een variant van de Kanban game die speciaal op testers is gericht en bieden die eenmalig aan als gratis Tutorial op het TestNet najaarsevent.

Je speelt de Kanban game hands-on, en vertaalt dit met je medespelers naar je dagelijkse praktijk. Een betere manier om er achter te komen wat lean en Kanban voor u en een slim testproces kunnen betekenen is er niet! ←



AS AGILE AS WE CAN BE

Door Rik Teuben • rteuben@vxcompany.com



Om écht agile te blijven moet je je je eigen 'agility' continue uit blijven dagen. Daarbij is improvisatietheater een belangrijk en doeltreffend en leuk hulpmiddel bij ons bedrijf gebleken. De uitgangspunten en afspraken die eraan ten grondslag liggen hebben veel raakvlakken met Agile in zijn puurste vorm.

Improvisatie

De meeste kennen improvisatietheater alleen van TV programma's zoals de Lama's, badgasten en het van recentere datum 'In goed gezelschap' dat de komende weken op Nederland 3 te zien is. Natuurlijk zijn dat professionele spelers, maar in essentie wijkt het niet af van de oorspronkelijke gedachten van Keith Johnstone -de geestelijk vader- en draait het altijd om samenwerken, focus houden, buiten kaders durven denken, falen als middel om te leren enz. Je hoeft niet bepaald een Agile Goeroe te zijn om te bedenken dat dat eveneens kernbegrippen bij Agile werken zijn. En dus gingen we met een aantal tester van VX Company aan de gang. Niet met het doel om het lolligste jongetje van de klas te worden, maar om te spelen met het begrip Agility. Het was een belangrijk uitgangspunt dat we de kwaliteit van het theaterspel in onze bijeenkomsten van ondergeschikt belang maakten ten opzichte van het doel om op een leuke maar zinvolle manier de begrippen flexibiliteit en wendbaarheid uit te dagen. Daar kwam nog eens bij dat allen ruime ervaring hadden met agile werken in de context van testen of software development, maar geen van ons verder was gekomen dan een bijrol in de schoolmusical of een sketch op het gouden bruiloftsfeest van tante Jo en ome Cor. We wilden onze eigen agility bestuderen. De vorm (improvisatietheater) was een hulpmiddel, en geen doel op zich. Het bleek een schot in de roos. Fun en effectief! Leerzaam en dikke pret. Agile zijn, en niet alleen agile doen!

Theatersport

Een belangrijk deel van het succes is toe te schrijven aan Martin Heynen van het bedrijf Zieffect.nl. Hij was gelukkig bereid om met geduld en toewijding onze bijeenkomsten te begeleiden. Martin is niet alleen een ervaren (bedrijfs)coach, talentvol blogger (martinheynen.blogspot.com) maar ook nog eens enthousiast beoefenaar van theatersport. Martin hielp ons over onze drempels heen ging samen met ons de uitdagingen aan. Het resultaat is te zien in zaal 11 en 12 om 19.30 op 2 oktober bij het TestNet najaarsevenement. ←

ABOUT TESTERS AND WASTE...

Door Dominic Maes • dominic.maes@quintes.be en Stefaan Luckermans • stefaan.luckermans@thaste.org



Toen we in 2010 terugkeerden van Eurostar in Kopenhagen, hadden we door de intense sneeuwval (en vertragingen op de luchthaven) voldoende tijd voor een discussie over de voorbije conferentie. De conferentie van dat jaar had een sterke 'Agile' inborst en er werd nogal vaak met de term LEAN getoverd tijdens verschillende presentaties. Aangezien we beiden betrokken waren bij een LEAN implementatie bij onze klant, vonden we het wachten al gauw wel een beetje 'Waste'. We besloten dan ook om onze tijd wat nuttiger in te vullen en begonnen een discussie over wat nu meer efficiënt is: Old-school gestructureerd testen of Agile testen. Wat volgt is een nuchtere zelfstudie over wat nu de meest efficiënte vorm van testen is.

Indien we willen onderzoeken of het testen binnen ons eigen IT project wel zo efficiënt is als we zelf zouden willen, kunnen we binnen de LEAN context wel hulp vinden. Wat volgt, is een kritische kijk naar de verschillende vormen van Waste binnen een project. We stellen ons hierbij de vraag of de Agile manier van testen efficiënter is dan de meer Traditionele vorm van testen.

Waste

Iedereen produceert afval of 'waste', ook in een IT-project ontsnap je er niet aan. Waste betekent hier uiteindelijk zoveel als tijd-, geld- of kwaliteitsverlies. Om te weten of je efficiënt bezig bent, is het wel nodig te weten of je niet onderhevig bent aan één van de verschillende vormen van Waste. In LEAN worden deze basisvormen aangeduid met het acroniem TIMWOOD++. Elke letter staat voor een andere vorm van Waste. Samengevat: Transport, Inventory, Moving, Waiting, Overprocessing, Overproduction, Defects, Talent and Bureaucracy. We zullen nu elke vorm van Waste kort doorlopen.

Transport

Iedereen kan zich wel iets indenken bij het woord transport. In meeste gevallen gaat het om het vervoer van personen of zaken van punt A naar punt B. Ook in een IT-project speelt dit mee. In een project gaat het vooral om je ideeën en kennis over te brengen (bijvoorbeeld via mails) om daarna het nodige budget los te peuteren, om je ideeën om te zetten of om contact te houden met je offshore partner. Werken er veel mensen aan je project, dan zal je transport exponentieel toenemen. Ook afstand tussen mensen (locatie) bemoeilijkt het transport. Kortom, hoe meer transport je hebt, hoe meer kans er is dat je Waste gaat aanmaken.

Inventory

Inventory of inventaris kan je helpen om een betere kijk te krijgen op het aantal deliverables dat in een project aangemaakt (moeten) worden. Denken we maar aan use cases, business requirements, user requirements, system requirements, test reports, progress reports, enzovoort. Een nachtmerrie om alles up-to-date te houden, →

te laten reviewen en getekend te krijgen. Hoe meer documenten, hoe meer noodzaak om alles onder controle te houden en dat kost tijd en geld.

Moving

Toename van transport, inventory, de hoeveelheid deliverables of informatie kan je letterlijk laten rondlopen (moving around). Afgewerkte producten of informatie komen nu eenmaal niet uit de lucht vallen. Vaak betekent het vragen, bedelen of stalken van verschillende personen of groepen om datgene te verkrijgen wat je nodig hebt in je project. Het verschil tussen transport en moving zit hem in datgene wat je verplaatst. Bij moving is dat niets. Enkel rondlopen van punt A naar punt B om de juiste mensen of dingen te vinden die je project in de goede richting duwen.



Waiting

Een vorm van Waste waar we niet veel dieper op in moeten gaan. Een project zit vaak vol wachten op een antwoord of een actie van een bepaalde persoon of dienst. Vaak wachten we *nét* op die ene overbezette persoon.

Overprocessing

Het nodeloos terugvallen op meer en meer processen. Het is niet de zoveelste review-ronde die je deliverable meer kwaliteit zal geven. Te veel processen zullen ook weerstand oproepen bij het projectteam. Een al zwaar overwerkt team zal niet juichen met wat extra werklast zonder de oorzaak van de problemen aan te pakken. De bedoeling is dat men de processen binnen een project zo simpel mogelijk houdt. Zit men *één*maal met een aantal processen, raakt men er moeilijk van af. En toch zijn processen nodig, denken we maar aan bijvoorbeeld verkeersregels. Te veel regels maken echter alles verwarrend. *Eén* gouden regel om te onthouden: 'Simplify, but don't make things simpler than its most simple form.'

Overproduction

Teveel trachten te bereiken dan eigenlijk nodig is in *één* keer is niet ideaal. Als men zich te veel toelegt op het produceren, in plaats van het opleveren van kwaliteit, zal dat resulteren in een project waarbij de scope verkleind of herzien moet worden.

Defects

We moeten in ons testtraject op zoek gaan naar bevindingen. Als we te veel bevindingen ontdekken, zullen we veel tijd verliezen in het beheer van die bevindingen, het oplossen ervan en het hertesten. Belangrijk is om jezelf de vraag te blijven stellen: 'Kunnen we ons product niet naar productie zetten met enkele bevindingen er nog in?' Niet alle bevindingen vragen om een (onmiddellijke) oplossing.

Talent

Talent of het ontbreken ervan is niet het enige belangrijke in deze vorm van Waste. Ook het laten groeien van het talent door middel van opleidingen en kennisoverdracht is belangrijk. Iedereen is het wel eens over het feit dat er tijd en geld verloren wordt met het inschakelen van niet gekwalificeerde medewerkers in een team. Er moet →

echter ook continu geïnvesteerd worden in de medewerkers, zodat het talent dat ze hebben ook nog eens efficiënt ontplooid wordt.

Bureaucracy

Een teveel aan papierwerk is nooit goed voor een project. Je kunt je project letterlijk laten verdrinken in papierwerk. Als de berg papier boven je hoofd groeit, is het moeilijk om te zien in welke richting je project evolueert. Papierwerk kan je project letterlijk bevriezen. Nochtans kan geen enkel project zonder papierwerk. Denken we maar aan toegangsrechten, release-notes, bevindingenbeheer, enzovoort.

De vormen van Waste die je terugvindt in een IT-project, zijn meestal ook combinaties. Denken we bijvoorbeeld maar aan het beslissingsproces in een klein bedrijf of in een groot bedrijf. In een klein bedrijf gaat dit meestal over een vlakke hiërarchische structuur waar het management in een meeting onmiddellijk kan beslissen (alle personen zijn namelijk aanwezig). Bij een groot bedrijf moet er meestal een proposal opgemaakt worden zodat de managers weten waarover moet beslist worden. Vrije momenten in de verschillende agenda's zijn moeilijk te vinden. Beslissingen worden genomen door managers die niet in de meeting aanwezig zijn. Zij moeten op hun beurt op de hoogte gebracht worden door notulen, enzovoort. Meer procedures leiden tot meer inventory, meer waiting, meer bureaucracy, vul zelf maar aan. Wat wél duidelijk is, indien we alle vormen van Waste doorlopen; is dat Waste op zichzelf nog meer Waste aanmaakt.

Root cause analysis

Laat ons Waste als ongewenst beschouwen. Hoe kunnen we er dan vanaf geraken? Omdat efficiënt te kunnen doen, moeten we terugkeren naar de 'Root Cause'. We spreken hierbij van de 6 'M'-en. Laten we ze even kort opsommen: Men, Mother nature, Management, Material, Money, Methods. Elke vorm van Waste heeft zijn oorsprong in één van de 6 'M'-en hierboven. Het is aan te raden om ook in je eigen project even stil te staan bij de oorzaken van je Waste door middel van een Root Cause Analysis. Het zal je inzicht geven in de oorzaken van de verschillende Waste vormen.

Maar wees voorzichtig in het trachten op te lossen van 'Waste'. Denken we maar even terug aan de grote bankencrisis. Deze crisis trof wereldwijd verschillende economieën met paniek en angst als gevolg. Grote ondernemingen en zelfs overheden werden zwaar getroffen en de reactie luidde unaniem: 'dit nooit meer, we moeten onze banken beter controleren en banken moeten transparanter rapporteren'. Iedereen reageerde positief op deze voorstellen omdat het in lijn lag van de verwachtingen. Het lag in lijn met de gedachte dat men enkel het probleem kan verhelpen door de methodes aan te passen, door de manier van werken te veranderen. Maar is dit echt wel zo'n goede oplossing? ➔

Muda – Mura – Muri

De negen categorieën van Waste zoals in het eerste deel opgesomd zijn, zijn de oorspronkelijke basisvormen zoals die door Taiichi Ohno¹ werden vastgesteld. Hij noemde deze basisvormen 'Muda' (Japans voor afval of Waste). Bij het bekijken van mogelijke verbeteringen wordt er te vaak gekeken naar de 'Muda'-vormen, terwijl de verbetering op een dieper niveau behaald kan worden.¹



Volgens Ohno bestaat wat we kunnen afleiden van de Root Cause Analysis enkel omwille van 'Unevenness, Inconsistency and/or Irregularity'. Deze vormen noemde hij 'Mura' (Japans voor ongelijkheid, onregelmatig of het gebrek aan eenvormigheid). Maar ook hier vond hij dat er misschien nog meer algemene gronden waren.

Ohno kwam vervolgens op de Root Causes voor Mura: Unreasonableness, Absurdity, Blindness, Overburden en Excessive stress. Deze basis-oorzaken noemde hij 'Muri' (Japans voor onredelijkheid, onmogelijk, boven iemands krachten).

Afvalverwerking: het juiste evenwicht

Zoals eerder aangehaald komen we uit een situatie waar Waste andere Waste creëerde. Het oplossen van bureaucracy door bijvoorbeeld het aanpassen van de werkmethodes is één van de mogelijkheden, maar je elimineert één vorm van Waste door er een nieuwe in de plaats te zetten. Het is zoals je kamer opruimen door alle rommel onder je bed te schuiven. Niet echt een oplossing dus. Maar is het wel nodig om alle Waste te verwijderen? Zijn er geen andere opties?

Neem eens een voorbeeld aan je werk: verkies je liever een baan waar alles voorspelbaar is en zelfs een beetje saai dan een baan met stress? Is het niet zo dat tijd sneller lijkt te gaan als je een dringend probleem probeert op te lossen? Moeten we hier echt de 'stress' weghalen? Misschien moeten we het evenwicht proberen houden tussen stress en burden (last)

Of wat met Unreasonable/Reasonable (Onredelijk/redelijk)? Een flinke tijd terug zaten er dertig kompels vast diep in een mijn in Chili. Het was onredelijk om van de reddingswerkers te vragen een oplossing te vinden voor het redden van deze mijnwerkers. Het was tevens onredelijk om te vragen bijna één kilometer diep te boren om ze te bevrijden, maar toch deden ze het. Het was makkelijker geweest om niets te doen, maar de onredelijkheid daagde hen uit om een ongekende oplossing te vinden die de zaak kon klaren.

Is alle Waste ongewild? Neen! Tot op een bepaald niveau zal Waste werken als een katalysator binnen je project. Te veel Waste zal je project hinderen, maar te weinig Waste doet precies hetzelfde. Alles in je project moet evenwichtig zijn: Je hebt behoefte aan specificaties, maar niet té veel. Je moet bevindingen oplossen, maar juist genoeg. Als testers wilden we ons hier graag opwerpen als afval verwijderaars, maar dat is een taak die iedereen in een project treft, niet enkel de testers. →

¹ Ohno, Taiichi (1988), *Toyota Production System: Beyond Large-Scale Production*, Productivity Press.

Agile = Waste control?

Kunnen we Agile development en Agile testen gebruiken om onze Waste te beperken en onder controle te houden in een IT-project? Dat is mogelijk, maar wat heeft Agile werken dat de traditionele manier van testen niet heeft? Mogelijk kunnen de twaalf principes van Agile helpen. Zijn deze twaalf eenvoudige principes voldoende voor een sluitend 'Waste management'? Zijn ze misschien niet wat té eenvoudig? Mogelijk zijn ze wel zo eenvoudig dat ze te moeilijk zijn om te implementeren. Misschien is het dat wel wat Einstein bedoelde met 'make things as simple as possible', maar we mogen zeker niet vergeten dat er een tweede deel van Einsteins zin was: 'but don't make things simpler than that'.

We weten dat de twaalf principes van Agile gebaseerd zijn op het Agile Manifest. Het manifest begint als volgt: 'We verkiezen'...

Mensen en hun onderlinge interactie boven processen and tools

Projecten worden natuurlijk door mensen gedaan, maar is er ruimte voor individuen in een team? Natuurlijk zal een getalenteerd iemand je project ten goede komen, maar hij kan je project ook ophouden. Je kunt die persoon zijn eigen weg laten zoeken, maar kan dat zonder een leider in het team die de regels bepaalt? En zijn deze regels dan geen processen?

Werkende software boven alles omvattende documentatie

Het idee is simpel: laat ons meer moeite steken in het afwerken van software zodat hetgeen we opleveren op zijn minst (goed) werkt. Maar kunnen we in een ontwikkelproces wel zonder documentatie? Wat moet de gebruiker zonder documentatie? En is documentatie in het geheel overbodig?

Samenwerking met de klant boven contractonderhandelingen

'Goede afspraken maken goede vrienden' en die afspraken worden vastgelegd in een contract. Maar is het wel degelijk mogelijk om de klant op een actieve manier te laten meewerken in een project? Krijgen we dan niet te veel bemoeienissen? Weet de klant wel wat goed is voor hem? Is de klant wel objectief genoeg?

Inspelen op verandering boven het volgen van een plan

We moeten als projectteam kunnen inspelen op veranderende wensen van onze klant. Maar een klant die continu zijn mening verandert, is dat wel zo handig? En wat met planning? Is werken zonder enige vorm van planning wel zo efficiënt?

Agile over Traditional softwaredevelopment

Het minste wat we kunnen zeggen is dat Agile- en Traditioneel testen totaal verschillende aanpakken zijn. Blijkbaar kunnen beide vormen niet zonder elkaar. Een zekere overlap is nodig. Traditioneel testen heeft zeker zijn tekortkomingen. Maar Agile in zijn extreme vorm heeft ook tekortkomingen. ➔

Een project moet voornamelijk 'in evenwicht zijn'. En misschien moeten we dat ook reflecteren in een (aangepast) Agile Manifest:



Mensen en hun onderlinge interactie in evenwicht met processen and tools

Werkende software in evenwicht met verstaanbare documentatie

Samenwerking met de klant in evenwicht met contractonderhandelingen

Inspelen op verandering in evenwicht met het volgen van een plan

Besluit

Kunnen we concluderen dat 'Traditioneel testen - Waste = Agile Testen'? Agile kan zeker helpen om efficiënter te werken, maar in zijn extreemste vorm verandert het eveneens één vorm van Waste in een andere. Agile in zijn extreemste vorm zal zijn eigen specifieke Waste creëren en daarom kunnen we beter spreken van 'Traditioneel testen - (Traditionele) Waste = Agile Testen - (Agile) Waste'.

(Met dank aan Rik Marselis voor de foto) ←

HELP! MIJN ORGANISATIE WIL AGILE...

Door Paul Custers • paul.custers@ns.nl • [@pcusters](https://twitter.com/pcusters)



Hoe kijken we over vijf jaar tegen Agile aan? Daar ben ik benieuwd naar. Ondanks dat het niet nieuw is, is Agile hot, momenteel misschien zelfs een hype. Scrum voert meestal de boventoon en alles gaat beter, projecten zijn altijd een succes. Waterval en RUP zijn termen uit een lang vervlogen verleden en sommigen lijken plezier te beleven aan het 'basen' van deze methoden. Levert Agile/Scrum over vijf jaar ook een vieze nasmaak op? Ik hoop van harte dat dit niet het geval is. Simpelweg omdat ik ook enthousiast ben over het toepassen van Agile en bij het toepassen van het framework positieve resultaten heb gezien en beleefd.

In zekere zin was ik een jaar of twaalf geleden al enthousiast over Agile in de vorm van de DSDM filosofie (Dynamic Systems Development Method – <http://www.dsdm.org/>), maar de link met Agile had ik aan het begin van mijn loopbaan nog niet gelegd.

Wat is dan nu de uitdaging of het probleem? In mijn ogen ligt dit vooral in het verhogen van de acceptatiegraad van Agile binnen organisaties. Om vanuit de energie die nu heerst een blijvend succes te maken van Agile.

Deze acceptatiegraad is per onderneming verschillend en de mate waarin Agile zuiver wordt toegepast ook. Soms werkt een deel van een ontwikkelteam volgens Scrum, bij andere organisaties werken meerdere teams compleet Agile en tot slot komt Scrummerfall nog wel eens voor. Het laatste geval waarbij teams watervalachtig werken, maar elementen als de daily stand-up gebruiken of een actieve Product Owner hebben is wellicht niet ideaal. Maar het toegenomen enthousiasme helpt desondanks vaak om succesvoller projecten uit te voeren.

Enkele voorbeelden

Wat kun je in het kader van die acceptatiegraad vanuit je testafdeling concreet betekenen in de promotie van Agile? Hieronder volgen enkele voorbeelden, waarvan sommige al succes hebben opgeleverd binnen mijn eigen organisatie.

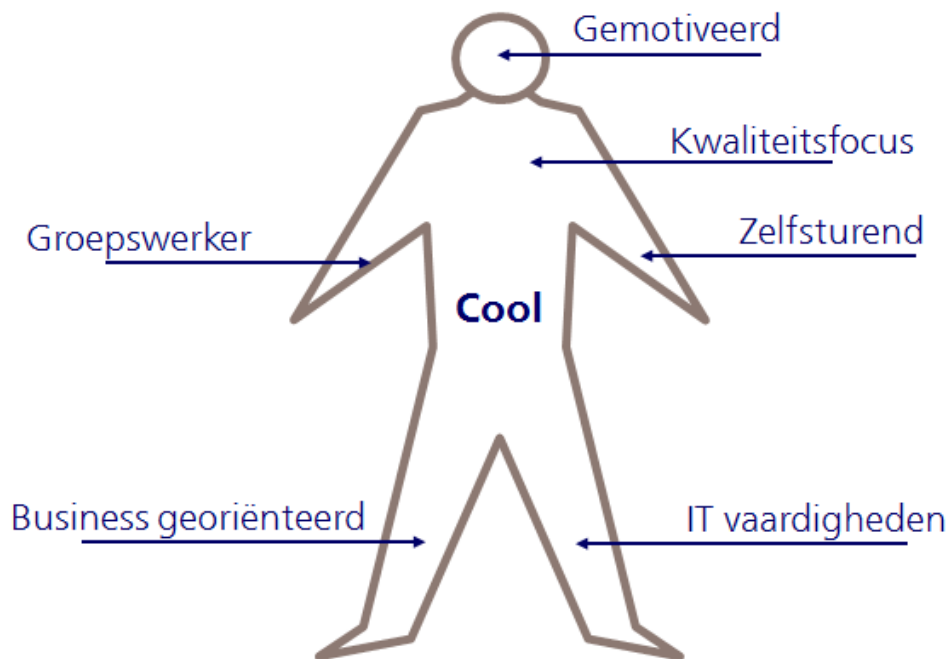
- *Dogfooding*. Je doet niet van de een op de andere dag Agile. Dat vereist naast training ook oefening. Zorg dat de teams die projecten binnen de eigen IT-afdeling oppakken het project Agile benaderen. Hierbij kun je eenvoudiger de gelegenheid creëren om te leren en fouten te maken. Op die manier voel je dat je (eind)klant direct hinder ondervindt als het een keer niet optimaal verloopt.
- Heeft je organisatie reeds een uitvoerbaar en goedgekeurd testbeleid? Agile hierin opnemen is een must.
- Creëer een *community* (Yammer, themamiddagen, bijeenkomsten voor testmanagers). Bespreek de best practices / lessons learned.
- *Richt resourceplanning van testprofessionals in*. Hiermee wordt kennisverlies beperkt. Bijvoorbeeld bij inhuur van externe medewerkers. Medewerkers met Agile ervaring wil je graag op nieuwe trajecten inzetten om de positieve lijn vast te houden.
- Testprofessionals hebben natuurlijk Agile trainingen gevolgd. Bijvoorbeeld Certified Scrummaster training.
- Zorg voor *gecreëerde testawareness* in de omgeving waarmee je afdeling te maken heeft. Dit is in de praktijk vaak 'de business'. →

- Neem je collega's uit de business mee naar *events op het scheidingsvlak van IT en business*. Samen een event bezoeken is behalve gezellig ook verhelderend voor beide partijen. Discussies volgen vanzelfsprekend na een leuke avond of dag.
- Deze laatste bullet is bestemd voor overige goede *suggesties*. De beste suggestie krijgt van mij een leuke NS-goodie ... mail me op bovenstaand e-mailadres!

Met bovenstaande punten is Agile vanuit een stevige basis te promoten. De Agile testprofessional speelt daar een actieve rol in. Dat vereist wel veel van deze persoon.

Agile tester

De ideale Agile testprofessional is aan de hand van het bijgaande figuur te duiden. Hij of zij is trots om in een team te werken, weet van aanpakken en gaat discussie niet uit de weg om het succes van de samenwerking te vergroten. De Agile testprofessional is op de hoogte van IT-ontwikkelingen en de gangbare methoden & technieken, heeft kwaliteitsgezindheid in zijn genen zitten en is in staat om voor de spreekwoordelijke hete vuren te staan en daarbij het overzicht te houden (cool te blijven). Niet onbelangrijk: hij of zij weet ook de relatie met de business te leggen en te onderhouden. Dat kan zowel vanuit de inhoud als het proces. In staat zijn om een linking pin te zijn is essentieel en vergt empathie.



Is voorgaande dan helemaal nieuw? Gelden deze eigenschappen niet voor de 'non-Agile test-pro'? Natuurlijk gelden dit soort eigenschappen in de Wereld Vóór Agile ook. Echter, bij meer traditionele ontwikkeltrajecten is verstoppen soms makkelijker. Daarmee doel ik niet zozeer op het bewust verstoppen en de rit uitzitten, maar voornamelijk op het in je comfortzone blijven. Hierbij verricht je je werkzaamheden prima, maar behaal je nooit de toegevoegde waarde die je zou kunnen bieden. Door de perfecte teamgrootte bij een Scrum-traject komen automatisch momenten dat een teamlid uit zijn of haar comfortzone moet komen én het ook eerder zal durven. Ideaal dus. Opvallend vind ik wel dat relatief veel Scrummasters een achtergrond als teamleider bouw hebben. Laten we hier als testprofessionals mooie kansen liggen? Toch nog in de comfortzone? →

Is nu alleen de meer seniore/ervaren testprofessional geschikt voor Agile/Scrum? Lijkt mij persoonlijk zeker niet het geval. Beginnende IT'ers brengen vaak een stuk energie en enthousiasme met zich mee. Dat ik niet zou willen missen in een project. Met hun beperkte ervaring stellen ze vaak de 'domme' vragen waardoor de doorgewinterde projectleden worden uitgedaagd in hun denkpatronen. De crux zit hem in het vinden van de goede balans in een team. Sorry, een open deur; eentje om met veel plezier op te pakken.

TMap en Agile passen niet samen?

Testwerkzaamheden binnen organisaties vinden vaak al langere tijd plaats op basis van TMap. De waarde hiervan wil je niet zomaar verliezen. Hier en daar vallen wel eens geluiden te bespeuren over starheid van TMap en dat TMap zou botsen met Agile trajecten. Correct me if I am wrong, maar TMap staat toch nog steeds voor Test Management approach? Ook na de toevoeging 'NEXT' enige tijd geleden? TMap mag voor sommigen 'de testbijbel' zijn, maar gezond en logisch nadenken, kritisch zijn en best practices toepassen is niet nieuw. Dat maakt in mijn ogen TMap méér dan geschikt om toe te passen in Agile trajecten. Zelf heb ik goede ervaringen met de Product Risico Analyse. Aan dit onderwerp is een uitgebreid hoofdstuk in het laatste TMap boek gewijd. De bij onze onderneming gebruikte tooling/methode Prisma levert als basis voor releases/sprintplanningen keurig input voor scoping-discussies en helpt bij de prioritering van de product backlog. Daarnaast 'hoeft' in een Agile traject niet alles 'exploratory' getest te worden. TMap biedt nog steeds een uitstekend overzicht van testtechnieken die prima toepasbaar zijn binnen een Agile traject. Mijn vermoeden is dat enige onbekendheid met TMap af en toe ook een rol speelt bij de beoordeling van de vermeende starheid. Op dit vlak kan de Agile testprofessional zich uitleven en zijn kennis laten spreken. Met als gevolg dat over vijf jaar Agile en TMap op geïntegreerde wijze furore maken. ←

AGILE TESTERS WISSELEN BEST PRACTICES EN RANDVOORWAARDEN UIT

Door Sanne Muskens • Sanne.Muskens@bartosz.nl mmv Beery Kersten, John Kronenberg, Jos van Rooyen en Petra Heesink

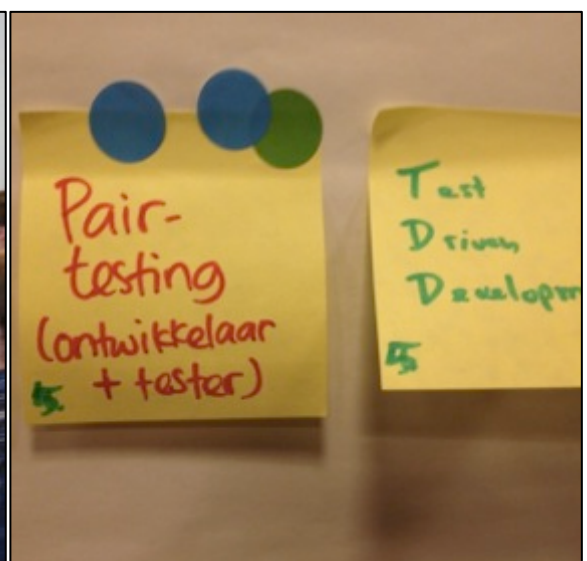


Tijdens de TestNet Summer School heeft Bartosz ICT een interactieve sessie georganiseerd met als titel 'Test Scrum of Scrums', waaraan 31 ervaren Agile testers hebben deelgenomen. Het doel van deze sessie was het uitwisselen van ervaringen op het gebied van Agile testen. Aanleiding voor het organiseren van de sessie is de inmiddels grote populatie Agile testers die tijdens het uitoefenen van hun vak dagelijks tegen dezelfde uitdagingen en vraagstukken aanlopen.

Om deze uitdagingen aan te pakken zijn Agile testers in de regel niet op zoek naar nieuwe theoretische inzichten, maar vooral naar hoe anderen in vergelijkbare situaties omgaan met het testen. Zij zijn op zoek naar ervaringen rondom Agile testen. Wat was de aanpak en het proces van de sessie? En wat waren de meest relevante uitkomsten van de sessie, gebaseerd op de uitspraken van de 31 deelnemers?

Agile

Agile krijgt veel aandacht en is tegenwoordig niet meer weg te denken in de IT. We merken dat onze klanten overstappen naar een op Agile manier van software ontwikkelen, maar dat de overgang van de traditionele Waterval-methode naar Agile voor uitdagingen zorgt. Met name de testdiscipline loopt tegen uitdagingen aan bij het uitvoeren van de werkzaamheden in een Agile omgeving. Voorbeelden zijn het testen met een incomplete testbasis, het ontbreken van entry-exit criteria, het toenemend belang van geautomatiseerde regressietesten, de aansluiting van het testen binnen het Agile team en de niet-Agile buitenwereld en de wijze en mate van documenteren van zowel testbasis als testproducten. Voor deze en andere uitdagingen vindt iedere organisatie haar eigen oplossing om hier lering uit te trekken. Hierdoor ontstaat de behoefte bij Agile testers om ervaringen uit te wisselen en best practices en randvoorwaarden met elkaar te delen op het gebied van Agile testen. In dat kader hebben we, onder leiding van Hans Ruesink, Beery Kersten, John Kronenberg, Jos van Rooyen en Petra Heesink, tijdens de TestNet Summer School de sessie 'Test Scrum of Scrums' georganiseerd.



→

Interactieve sessie 'Test Scrum of Scrums'

Tijdens de sessie hebben 31 Agile testers ervaringen uitgewisseld met betrekking tot Agile testen. Er stonden twee vragen centraal, namelijk: 'Hoe pas je testtools effectief toe in een Agile omgeving?' en 'Hoe integreer testsoorten wanneer in een Agile project moet worden samengewerkt met een niet-Agile project?' Samen met de deelnemers is per vraag allereerst een kader gecreëerd van verschillende testsoorten dan wel testtools binnen een Agile omgeving. Welke testsoorten en testtools kom je tegen bij Agile testen en waar heb je ervaring mee? Vervolgens is aan alle testers gevraagd om vanuit hun eigen Agile testervaring voor de onderwerpen testtools en testsoorten best practices met bijbehorende randvoorwaarden te definiëren. Een best practice is een techniek, werkmethode, proces of activiteit die zich als effectiever heeft bewezen dan enige andere techniek, methode enzovoort. De gedachte hierbij is dat met de juiste werkmethode het werk uitgevoerd kan worden met minder problemen, onvoorziene complicaties, inspanning en met betere eindresultaten. Het is dus voor organisaties belangrijk de best practice binnen hun eigen domein te kennen en de eigen manier van werken hiermee te kunnen vergelijken. De bijbehorende randvoorwaarden stellen eisen aan het Agile project die vanuit het project niet kunnen worden beïnvloed, maar wel nodig zijn om volgens de best practice te kunnen werken. Na het definiëren van best practices en bijbehorende randvoorwaarden zijn deze besproken met de deelnemers en toegelicht aan de hand van voorbeelden. Vervolgens zijn ze door de ervaren Agile testers geprioriteerd door middel van 'dot voting'. Iedere deelnemer mocht met twee stickers aangeven welke beste practice en welke randvoorwaarde hij of zij nu als meest relevant beschouwt.

De onderstaande opsomming geeft de zeven meest relevante best practices en randvoorwaarden uit de sessie weer. Deze zijn dus gezamenlijk gedefinieerd en gekozen door alle 31 Agile testers.



Best Practices

Watervalervaringen

Eén van de belangrijkste en meest bruikbare best practices betreft het niet over boord gooien van opgedane kennis over testsoorten. Vind het wiel bij Agile testen niet opnieuw uit, maar maak gebruik van je ervaringen met testsoorten uit Watervalprojecten. Belangrijke vraag hierbij is of de verantwoordelijkheid van de testsoort binnen of buiten het Agile team ligt. Bij de unit- en unitintegratietest is dit duidelijk binnen het Agile team, maar bij de systeem- en acceptatietests is er een aantal mogelijkheden. Voor deze testsoorten is het mogelijk een afzonderlijk testteam buiten het Agile team of zelfs buiten de organisatie in te richten. Ook wordt er meestal voor gekozen om de productieacceptatietest buiten het projectteam uit te voeren. Deze test vereist specifieke (systeembeheer) expertise van de tester plus een testomgeving die zo productiegetrouw mogelijk is. →

Beide zijn vaak maar beperkt in het Agile team aanwezig. Ditzelfde geldt voor het toepassen van Productrisicoanalyses (PRA). PRA is een bewezen methode die ook in een Agile omgeving kan worden toegepast om de teststrategie te bepalen. Als tester moet je er zorg voor dragen dat de testactiviteiten worden gerelateerd aan de risico's die zijn verbonden aan het op te leveren product of de functionaliteit binnen een sprint. Dit kan door middel van het houden van een high level PRA-sessie aan het begin van het project. Nadeel hiervan is dat gebruikers hun risico-inschatting geven op basis van de verwachte eindrelease, waardoor het moeilijker is om een teststrategie te bepalen voor een tussenrelease van één sprint. Om deze reden kan het houden van een mini PRA-sessie aan het begin van elke sprint nuttig zijn.

Betrek stakeholders

Een andere belangrijke best practice is de betrokkenheid van de stakeholders en acceptanten. De producteigenaar vertegenwoordigt de behoeften en wensen van de stakeholders aan een Agile team. In Scrum is de 'Product Owner' één van de kernrollen. Hoewel de Product Owner formeel de stakeholders vertegenwoordigt, is het aan te raden om de stakeholders wel te betrekken in de communicatie. De stakeholders moeten het systeem uiteindelijk accepteren en je wilt niet het risico lopen dat er niet aan de verwachtingen wordt voldaan. Door betrokkenheid te creëren bij stakeholders kun je meer en eerder feedback ontvangen waardoor dit risico vermindert. Te weinig betrokkenheid zorgt voor meer vragen in het team en minder feedback. Houd er rekening mee dat stakeholders ook andere projecten en activiteiten hebben. Betrek de stakeholders daarom op een efficiënte en effectieve manier bij het project. Enkele voorbeelden om dit te doen zijn: door het organiseren van een kick-off voor start uitvoer project. Tijdens de kick-off krijgen de stakeholders de gelegenheid hun verwachtingen te uiten. Een andere manier is het organiseren van periodieke formele afstemmomenten. Dit moet echter wel via de Product Owner geregeld worden. Maak de stakeholders enthousiast en betrek de stakeholders verder bij het traject door ze de applicatie te demonstreren na iedere sprint. Stuur na iedere sprint voortgangsrapporten, breng ze op de hoogte van wat er in de huidige sprint gedaan is en wat er in de Backlog zit.

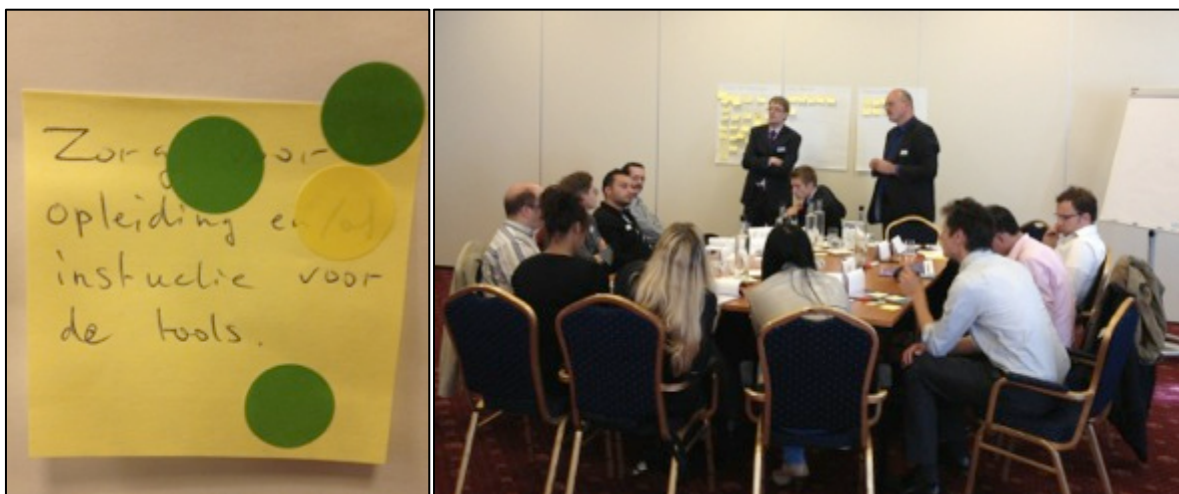
Vroeg starten met testen

Start zo vroeg mogelijk met testwerkzaamheden. Door vroeg in de sprint te beginnen met testen, heb je de mogelijkheid om tussentijds nog bij te sturen. Dit betekent dat zodra een testbare eenheid gereed is, je als tester direct aan de slag moet kunnen. Ongeacht of de testbasis compleet is of niet. In een Agile omgeving is de testbasis aan het begin van de sprint niet 100% compleet en wordt deze gedurende de iteratie gecompleteerd. Hierbij is een nauwe samenwerking tussen ontwerper, ontwikkelaar en tester aan te raden, want hierdoor raak je als tester eerder betrokken bij ontwerp en ontwikkeling van het deelproduct. Vroegtijdig kennis nemen van 'wat' en 'hoe' iets gebouwd wordt, is nuttig voor het opstellen van testgevallen en zorgt dat je als tester minder afhankelijk bent van een testbasis. Het is dus slim om de tester al in een zo vroeg mogelijk stadium te betrekken, het liefst al tijdens de design- en bouwfase. Met name als een taak erg technisch is kan dit nuttig zijn. Testen betekent niet altijd fysieke testuitvoering, het kan ook valideren en/of toetsen betekenen. Verder is snelle feedback essentieel voor het vroeg kunnen starten met testwerkzaamheden, een proactief team informeert de tester direct als iets af is.

Pair Testing

Een andere best practice is Pair Testing. Pair testing is een softwareontwikkeltechniek waarbij twee teamleden samenwerken op één toetsenbord om de softwareapplicatie te testen. Hierbij kan gedacht worden aan een →

combinatie van de ontwikkelaar en tester en/of aan een tester en ontwerper. Pair testing is voornamelijk aan te raden bij complexe taken. Wanneer een User Story of taak complex of high risk is, dan is het slim om dit samen te testen en het zogenoemde 4-ogen principe toe te passen. Een ontwikkelaar en tester kunnen besluiten om samen de testvoorbereiding en/of -uitvoering te doen. Door de nauwe samenwerking verbetert de kwaliteit van de op te leveren software. Indien er binnen de Agile omgeving volgens Test-Driven Development (TDD) wordt gewerkt, kunnen ontwikkelaar en tester samen tests bedenken. TDD is een software ontwikkelmethode waarbij eerst tests worden geschreven en daarna pas de code. Randvoorwaarde hierbij is wel dat de tester de broncode kan lezen. Het profiel van een (goede) Agile tester omvat daarom ook technische kennis en skills. Het is belangrijk dat zij weten hoe de code werkt en geschreven wordt. Dit helpt de Agile tester beter samen te werken met de ontwikkelaar, met name omdat de communicatie beter verloopt. Een voorbeeld is het beschikken over basiskennis JAVA of Microsoft.Net. Een Agile tester kan hierdoor helpen om Unit Test scripts op te stellen of te reviewen, en kan beter sparren in gesprekken over bijvoorbeeld Unit Test dekingsgraad. Maar ook tijdens het poken van de User Stories is het belangrijk voor een tester dat hij/zij inzicht heeft in het werk van een ontwikkelaar. Je pokert namelijk voor het hele team, inclusief ontwerp en ontwikkeling, en niet alleen voor je eigen discipline.



Randvoorwaarden

Communicatie en transparantie

Naast de broncode kunnen lezen en het hebben van technische kennis en skills, is het communicatief vaardig zijn, benoemd door de hele groep Agile Testers als een belangrijke randvoorwaarde. Als Agile tester moet je in een Agile omgeving nauw kunnen samenwerken met ontwerpers en ontwikkelaars. Bovendien moet je als Agile tester kunnen schakelen met de verschillende disciplines. Binnen het Agile manifeste heeft goede communicatie binnen het team niet voor niets een prominente plaats. Daarnaast is transparantie van belang. In een Agile omgeving wordt minder vastgelegd in specificaties en eerder gestart met ontwikkelen. Waardoor de kans bestaat dat specificaties achterlopen op de huidige ontwikkeling. Een ongewenste situatie voor een tester. In dit geval is transparantie over je testwerkzaamheden belangrijk. Maak duidelijk aan anderen 'hoe' en 'waarom' je een testactiviteit uitvoert. Beperk je daarbij niet tot alleen het Agile team, maar zoek ook contact met de stakeholders. Maak bijvoorbeeld een presentatie van je mastertestplan en betrek de stakeholders bij de demo of tussentijds in de sprint. Houd hen op de hoogte, dan houden zij jou ook op de hoogte. Met als resultaat een (meer) zelfsturend team en stakeholders. Communiceer ook afspraken die je binnen een team maakt en die invloed hebben op de stakeholders met betrekking tot de Backlog en User Stories specifiek. →

Belang Product Owner

De rol van de Product Owner is belangrijk in Agile projecten. Essentieel voor deze rol is het hebben van daadwerkelijk eigenaarschap, het hebben van business- en domeinkennis en mandaat. Dit is een randvoorwaarde voor het succes van Agile testtrajecten. De Product Owner is het aanspreekpunt van het Agile team in het geval van functionele onduidelijkheden. Hij of zij wordt geacht om snel knopen door te hakken en/of vragen te beantwoorden vanwege de korte iteratie, namelijk een sprint van 1-4 weken. Daarnaast vertegenwoordigt de Product Owner de belangen van de stakeholders. Als de Product Owner zijn rol niet goed vervult heeft de tester daar direct hinder van. Voor de tester is het van belang om de Product Owner te zien als de collega met domein- /businesskennis, als de beslisser in het geval van functionele onduidelijkheden en als belangrijk(st)e acceptant. Van een Product Owner wordt verwacht dat hij of zij proactief handelt. In een Agile team moeten belangrijke beslissingen soms ad-hoc worden genomen. De Product Owner moet dan direct beschikbaar zijn en eventueel een besluit kunnen nemen. Om een (deel)oplevering te kunnen accepteren zijn acceptatiecriteria en/of requirements nodig. Als Agile tester kun je de Product Owner ondersteunen bij het opstellen en de formulering van acceptatiecriteria en/of requirements. Dit verkleint de kans op interpretatieverschillen en conflicten.

Eigenschappen Agile tester

Gedurende de sessie is naar voren gekomen dat veel andere randvoorwaarden voor een succesvol testtraject te maken hebben met de eigenschappen van een tester. Samengevat zijn de volgende randvoorwaardelijke eigenschappen van een Agile tester genoemd door de deelnemers: een positief kritische houding; senioriteit; Agile kennis; in teamverband moeten kunnen werken; enkele jaren testervaring; 'out-of-the-boxdenken'; communicatief vaardig; technische kennis; broncode kunnen lezen. Een aantal van deze eigenschappen zijn in dit artikel al toegevoegd, zoals communicatief vaardig zijn, het hebben van technische kennis en het kunnen lezen van de broncode. Graag willen we nog stilstaan bij de positief kritische houding. Als Agile tester moet je namelijk (positief) kritisch zijn en blijven. Bij het werken in teams is een mogelijke valkuil om mee te gaan met de rest. Als tester doe je geen aannames en dat moet je als Agile tester ook niet doen, we nemen niet alles voor waar aan. Een positief kritische houding, inlevingsvermogen en out-of-the-boxdenken zijn hierbij belangrijk. Verder wordt de senioriteit van de tester genoemd als randvoorwaarde en belangrijke eigenschap. Senioriteit is vereist binnen het team om soepel Agile te werken. Een Agile tester moet in ieder geval weten wat de spelregels van Agile zijn, teamgevoel en enkele jaren testervaring hebben. Agile testers hebben veel ervaring nodig en eerder opgedane kennis in Watervaltrajecten moeten zij aanwenden in de Agile trajecten. Het is geen kwestie van het wiel opnieuw uitvinden. Kennis en ervaring moeten op de juiste manier ingezet worden. Tegelijkertijd wordt gewezen op het vermogen om out of the box te denken, van gebaande paden te stappen en niet vast te houden aan de geldende standaarden en methoden. Dit wordt vaak gezien als een uitdaging voor een tester die een schat aan ervaring en kennis heeft. Agile vereist een bepaalde mindset en volwassenheid van de tester.

Het succes van ervaringen uitwisselen

In het kader van continue verbetering en kennisdeling vonden de 31 ervaren Agile testers het uitwisselen van ervaringen op deze manier leerzaam en nuttig. Organisaties kunnen met behulp van deze methode ervaringen uitwisselen over (Agile) projecten heen en hun eigen best practices definiëren. Op die manier wordt er meer bruikbare kennis opgedaan en kan men 'leren van elkaar'. ←

ERGSTE MISVATTINGEN OVER TESTEN BIJ AGILE

Door Jaap van der Leer • JaapVanDerLeer@Valori.nl



Agile ontwikkelen en testen: water en vuur. Over testen bij Agile hoor je de meest vreemde opvattingen. Er zijn zelfs mensen die denken dat Agile is uitgevonden om niet te hoeven testen. Het testen bij Agile is geïntegreerd. Het gaat niet om functies, het gaat om rollen. Geen opvolgende fasen, alles tegelijk. Aloude wijsheden uit testland vallen om.

Ik neem je mee op een ontdekkingsreis naar de echte waarde en het nut van testen bij contemporaine bouwtechnieken. We maken korte metten met de twaalf allerergste voorbeelden van misvattingen. Ze zijn afkomstig uit onze praktijkervaringen. Onderweg staan we stil bij wat do's en don'ts.

Geen requirements meer nodig

We zien al te vaak dat gedefinieerde testgevallen de plaats van requirements innemen. Begrijpelijk, niet de bedoeling: doel van testen is vaststellen dat aan requirements voldaan wordt. Het Agile ontwikkelteam heeft nog steeds eisen nodig, in de vorm van goed uitgewerkte user stories. Met duidelijke beschrijvingen van goed- en foutpaden. Team en Business samen. Goedgekeurd.

Testen is een aparte activiteit

Als je wilt genieten van de voordelen van Agile, dan moet testen een teamactiviteit zijn. Ook ontwikkelaars en gebruikers moeten eraan geloven.

De tester (of testcoördinator, testmanager, testgoeroe) houdt zich juist niet bezig met testen, hij/zij zorgt ervoor dat in het Agile team iedereen test. Dat vereist nieuwe skills. Een tester kan zich niet meer afzijdig houden van coding.

Bevindingen bijhouden is nutteloos

In goede Agile teams vindt meerdere keren per dag een daily build plaats. Deze daily build omvat ook elke keer een geautomatiseerde uitgevoerde testronde. Maar wat gebeurt als met een test een fout is ontdekt?

Al te makkelijk fikst het team de fout fluks. Het team heeft niet naar de oorzaak gekeken. De tester zal juist dan zijn toegevoegde waarde tonen en doorprikken naar de oorzaak. Zelfs als die ligt bij de aansturing van het project.

Testen doe je achteraf

Het team moet trachten alle testen uit te voeren binnen een sprint. Alle testsoorten tegelijk oppakken. Dat is in de praktijk heel moeilijk. Want verschillende testsoorten het vereisen diverse testomgevingen. Tot en met een omgeving voor de ketentest. Hoe beter het team hier in slaagt, hoe groter de opbrengst aan vroeg gevonden afwijkingen. →

Testen bouwt kwaliteit in

Kennen jullie dat? Testen is een visnet dat de bugs uit het systeem haalt zodat kwaliteit blijft. Al in 1994 werd dat ontkracht.

Het team bouwt kwaliteit in het systeem. Kwaliteit creëer je en het is niet wat achterblijft nadat alle bugs zijn verdwenen. Juist hier overstijgt de tester zichzelf. Door vanaf de kansel te prediken en op de barricaden te vechten voor het uitvoeren van de juiste aanpak.

Testen zonder de Business is goed te doen

Agile staat en valt met Business involvement. Dat beperkt zich niet tot de bouw!

Testen is intuïtief bij Agile

In de praktijk ziet men dat iedereen meent te kunnen testen. Het is een rol, geen functie. Maar de echt goede Agile trajecten waarderen een vakkundige tester. Een tester die met een sessie de risico's boven tafel krijgt en die weet hoe je daar het best mee kunt omgaan. Want testen is nog steeds een vak.

Changes: graag!

Agile: de vrijbrief om elke keer weer en meer wijzigingen in te schieten. Agile teams verwachten wijzigingen. Geeft niet hoe veel en geeft niet hoe vaak. Maar wat gebeurt als 'het management' halverwege vierentwintig changes inschiet? Zou de stabiliteit van wat je al hebt misschien beïnvloed worden?

Hier moet de tester (vanuit zijn functie, niet zijn rol) de rem zijn. Agile werkt goed als elke sprint een beperkt aantal features van hoge kwaliteit maakt. Changes moeten naar de volgende sprint gestuurd worden. Tenzij het absoluut niet anders kan. In dat geval moet de tester een stevig plan maken om risico's te mitigeren.

Definition of done: foutloos

Hoezo? Wie heeft dat gezegd? Een goede discussie over kwaliteit is een prima basis om te besluiten tot 'done'. Vergeet dan ook regressie effecten niet mee te nemen. Streven naar volledige afwezigheid van fouten gaat te ver. Erger is de afwezigheid van de testfunctie bij het besluit om een fout te accepteren. Helaas komt dat vaker voor dan we verwachten.

Testen is eigenaar van geautomatiseerd testen

100% fout. De tester moet dan de afweging maken tussen 'handmatig testen' versus 'test automatisering bouwen'.

Het is het beste idee om bouwers verantwoordelijk te laten zijn voor de geautomatiseerde tests van hun code. Dan kunnen ze ervoor zorgen dat hun [later gebouwde] programma voldoet aan de testen. Wat kan je beter hebben? Mits de tester natuurlijk zelf controleert dat de geautomatiseerde test de lading dekt, compleet is, de juiste dingen controleert en foutpaden verifieert. Zaken waarin de tester excelleert.

Negatief testen noodzaak

Wat mij gelijk brengt op mogelijk de meest wijdverbreide misvatting: in Agile hoef je alleen te testen wat moet werken. Eigenlijk is dat van ondergeschikt belang. Testen dat wat niet mag werken inderdaad niet werkt, ➔

heeft daadwerkelijk toegevoegde waarde. Dat je niet om beveiliging heen kan, te lange adressen niet kunnen, foutmeldingen begrijpelijk zijn. Een creatieve tester die vandaag denkt als crimineel en morgen als nieuwe gebruiker is goud waard.

AINO

Oftewel Agile in name only. Vaak gecombineerd met PINO.

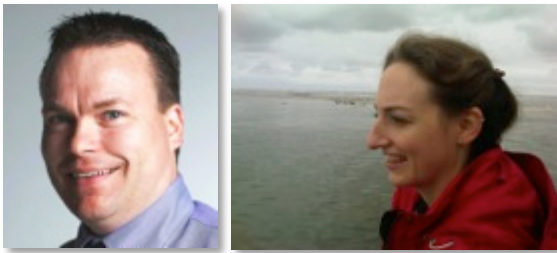
Wat testers van oudsher denken is volgordelijk testen: na oplevering nog een acceptatietest en dan een ketentest en dan een end-to-end test en een productieacceptatietest en een beheertest.

Als een tester dat doet, kun je beter watervalprojecten inrichten.

Hoe moeilijk ook, testen in Agile is anders. ←

KUN JE AGILE TESTEN LEREN IN EEN CURSUS?

Door Jurian van de Laar • jla@improveqs.nl en Manon Penning • mpe@improveqs.nl



Agile is actueel en we zien veel organisaties die overstappen op een Agile werkwijze. Hoe leer je dan als tester om je werk op een Agile manier in te richten? Bij het leren Agile Testen komt veel meer kijken dan theorie alleen. Samenwerken en communiceren zijn voor een Agile tester bij uitstek belangrijke vaardigheden.

Dat vraagt om een andere aanpak dan een boek of een traditionele cursus, maar welke dan?

Hoe leer je Agile testen?

Agile is actueel en we zien veel organisaties die overstappen op een Agile werkwijze. Maar hoe leer je om op een Agile manier te testen? Soms worden medewerkers door hun opdrachtgever naar een cursus gestuurd. Het is de vraag of dit de beste weg is. Agile benadrukt het belang van gemotiveerde individuen, dus zou de drijfveer om te leren niet meer uit de medewerker zelf moeten komen? Dan nog blijft de vraag wat de beste manier is. De theorie achter Agile is relatief eenvoudig. Het Agile manifesto en de twaalf principes vormen de basis. Over de invulling met Scrum, Kanban of op een andere manier is de theorie ook gemakkelijk te vinden. Het toepassen van deze theorie is echter een ander verhaal, zeker voor testers. Traditioneel kwamen testers al snel in de verdrukking wanneer andere projectfasen vertraagd raakten. In de typisch korte iteraties van een Agile ontwikkelproces lijkt het inpassen van alle testactiviteiten een nog grotere uitdaging te worden. Bovendien: hoe kan het beste omgegaan worden met de continu veranderende testbasis? Hoe doen we de risico-analyse in een Agile project? Wanneer testen we de niet-functionele kwaliteitskarakteristieken? Hoe overleef je als tester in een Agile team en hoe werk je goed samen met andere disciplines?

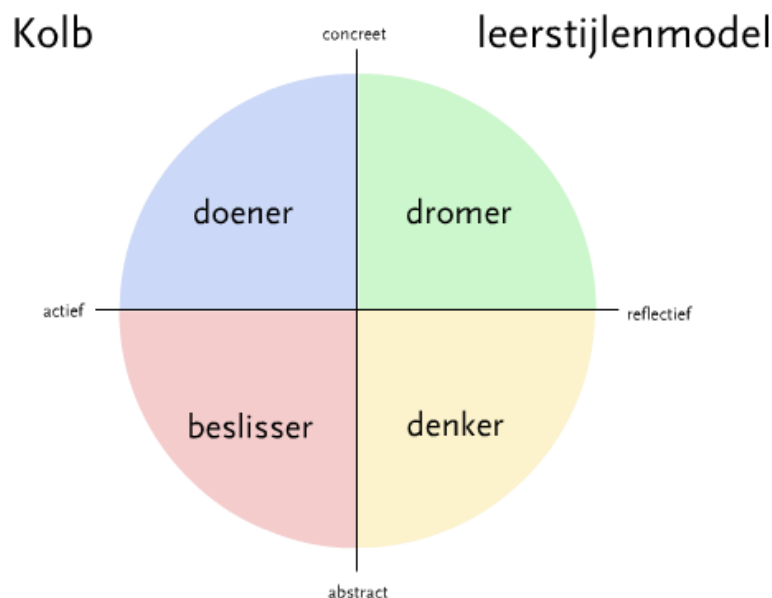
Er is al veel te vinden

Er zijn in de afgelopen jaren al diverse boeken verschenen die antwoorden geven op belangrijke vragen over Agile testen. Het meeste bekende boek is Agile Testing (Crispin & Gregory, 2008). Ook Test Driven Development (TDD) biedt oplossingen. Daarbij raken ontwikkeling en testen nauw met elkaar verweven, waardoor testen een onlosmakelijk deel van het ontwikkelproces wordt en altijd precies genoeg testen worden geschreven op een heel kortcyclische manier. Echter, TDD is een ontwikkelproces, geen testaanpak. Bovendien wordt niet het volledig spectrum van testsoorten afgedekt: TDD heeft betrekking op unit testen, ATDD (Acceptance Test Driven Development) gaat met name over acceptatietesten. Systeemtellen worden in deze Agile methodieken eigenlijk niet ingevuld. TDD is wel een uitstekend startpunt zijn om testen en testautomatisering op andere niveaus ook Agile in te richten. Een organisatie waarin TDD wordt toegepast, is zich al bewust van het belang en de noodzaak van testen. De ontwikkelaars zijn gewend aan het automatiseren van testen. Deze awareness is goed bruikbaar om ook testen en testautomatisering in te richten op andere niveaus. Ook zal de bereidwilligheid van ontwikkelaars om mee te helpen bij testautomatisering in deze organisaties hoger zijn. Testautomatisering is immers erg belangrijk binnen Agile. We kunnen dus concluderen dat er weliswaar meer en meer handvatten te vinden zijn, maar een →

éénduidige aanpak voor Agile testen hebben we daarmee nog niet gevonden. Ironisch genoeg zou zo'n aanpak waarschijnlijk ook niet flexibel genoeg zijn voor een Agile manier van werken!

Theorie is niet genoeg

Bij het leren Agile testen komt dus veel meer kijken dan theorie alleen, meer dan wat je uit een boek kunt halen of wat een traditionele cursus kan bieden. Samenwerken en communiceren zijn voor een Agile tester bij uitstek belangrijke vaardigheden. Dat vraagt om een andere aanpak dan een cursus in de klassieke vorm met de docent als 'zender' en de deelnemers als 'ontvangers'. Door nieuwe werkvormen krijgen trainingen een interactiever karakter en kunnen deze vaardigheden in cursusverband geleerd en geoefend worden. Dit beeld komt overeen met inzichten die onderwijskundig onderzoek hebben opgeleverd. Er wordt al jaren onderzoek gedaan naar de manier waarop mensen leren. Er zijn verschillende modellen ontwikkeld, waarvan het model van Kolb wellicht de bekendste is (Kolb & Fry, 1974). Kolb stelt dat er vier verschillende dominante leerstijlen zijn en dat iedere lerende een voorkeur heeft voor één van die vier, maar dat de optimale leerervaring pas plaatsvindt wanneer alle vier geraakt zijn.



Bron: dilemmamanager.nl

(© DilemmaManager B.V. 2011)

Toch zijn er diverse controversiële discussies gaande over de vraag of iedereen elke leerstijl wel in voldoende mate kan toepassen om effectief te zijn. Sterker nog, steeds vaker wordt in twijfel getrokken of iedereen wel alles kan leren (Lilienfeld et al, 2010). In de Agile wereld wordt vaak benadrukt dat Agile testen alleen gedaan kan worden door bovengemiddeld goede medewerkers (Marselis, 2009). Dat zou betekenen dat niet iedereen geschikt is als Agile tester, hoewel daarbij direct opgemerkt wordt dat training een noodzakelijke stap blijft om de rol van Agile tester te kunnen gaan vervullen (Tijman & Jimmink, 2008).

Leerstijlen en Agile

De Agile uitgangspunten gaan uit van de kracht van het individu en samenwerking. Dat wil niet zeggen dat iedereen hetzelfde moet kunnen, maar dat elk teamlid met zijn eigen unieke capaciteiten kan bijdragen aan het team. Diversiteit in disciplines en rollen in het team vullen elkaar aan. Dit principe zouden we kunnen toepassen →

in de vertaling van het leerstijlen model naar een Agile context. Misschien dat één bepaalde leerstijl het beste past bij de 'typische Agile tester', maar het is waarschijnlijker dat bij de verschillende rollen, bijvoorbeeld binnen een Scrum-team, ook verschillende leerstijlen passen. Zo kan iedereen in het team de best passende leerstijl (of leerstijlen) doorlopen voor zijn eigen optimale leerervaring.

Bij dromers en denkers past typisch een theoretische leerstijl. Zij zullen waarschijnlijk niet hun sterkte hebben in het nemen van snelle beslissingen tijdens een sprint, maar wel in het leggen van verbanden, het analyseren van user stories en het denkwerk bij het opstellen van een risicoanalyse of een teststrategie. Een beslisser, altijd op zoek naar voorbeelden van anderen (best practices) en delen van kennis, zou als product owner goed op zijn plek zijn. Doeners zijn eerder op zoek naar praktische werkvormen, zoals 'learning on the job' en bij uitstek geschikt om binnen het Scrum-team in de sprint aan de slag te gaan. Als de verschillende leerstijlen in een training aan bod komen, bijvoorbeeld door het Scrum proces te ervaren, kan voor het volledige team een optimale leerervaring plaatsvinden. Het rendement hiervan is sterk afhankelijk van het team zelf en van de Scrum Master, die als eigenaar van het Scrum-proces ook de leerervaring kan faciliteren en stimuleren.

Naast diversiteit in leerstijlen en de bijbehorende variatie in werkvormen, zijn er nog andere ontwikkelingen die de plaats innemen van het klassieke 'zender/ontvanger'-onderwijs. Nieuwe technologieën brengen allerlei mogelijkheden met zich mee. Met een interactieve online leeromgeving kan de cursusinhoud beter adaptief gemaakt worden en de locatie flexibel. Zeker in een Agile omgeving is het van belang om jezelf te blijven ontwikkelen. Veel informatie en ervaringen van anderen zijn beschikbaar op blogs, in (online) artikelen en op conferenties. In een dergelijke combinatie van training, online leeromgeving en zelfstudie is veel breder dan alleen een cursus in de klassieke zin. Het wordt een integraal leertraject waarin ook de coach past die op afstand de cursist of klant ondersteund (e-coaching).

De aanpak in de praktijk

In de afgelopen jaren hebben wij zelf veel van het bovenstaande geleerd en toegepast in het ontwikkelen en geven van Agile trainingen. Sinds vorig jaar hebben we ook ervaring opgedaan met een nieuwe training, die ook goed past bij onze visie op opleiden, namelijk de training Certified Agile Tester (CAT). Dit is een internationale training die op dit moment al door zes Nederlandse aanbieders wordt aangeboden en door een aantal andere Europese training providers.

Deze training bevat naast theorie ook veel praktijk, waaronder het doorlopen van het volledige Scrum-proces. Daarbij speelt samenwerking en team dynamiek een cruciale rol. Voor dit artikel hebben we alle cursisten uit de eerste groep nog eens benaderd om te toetsen of deze training hen ook echt geholpen heeft om Agile testen te leren en of ze baat hebben gehad bij deze training om hun werk beter te kunnen doen. Dit is een samenvatting van hun reacties: →

Patrick: 'Vooraf dacht ik 'alles wel te weten' omdat ik net negen maanden in een Scrum-team had gewerkt. Desondanks heb ik toch nieuwe dingen geleerd en in de bekende dingen meer de theorie en diepgang gehoord. Ik weet graag het hoe en waarom van dingen waar ik mee bezig ben. Dus de theorie achter de praktijk. Daar heeft de cursus me heel goed bij geholpen.'

Heeft deze cursus iets veranderd in jouw werk?

Cees-Jan: 'Meer exploratory system testen en de session sheets zijn een blijvertje. Ook is er meer aandacht voor risk-based testing.'

Marco: 'Ik heb ook een aantal punten uit de discussies en oefeningen mee kunnen nemen in mijn projecten. Ik vond het een zeer nuttige cursus. Vooral door de interactieve opzet, de afwisseling van theorie, praktijk en discussies. De opdrachten benaderden het Agile werken goed. Ook de indeling van alles in één week was denk ik goed voor de groepsvorming. Ik denk dat je voor de rest veel moet leren/ondervinden door het gewoon te doen. Ik zou daarbij wel aanraden om een expert in te huren die je de eerste periode kan ondersteunen. Zie ook de blog die ik in februari over deze cursus geschreven heb (Bunt, 2012).'

Vind je dat deze cursus een goed manier is geweest voor jou om Agile testen te leren?

Niko: 'De basis is goed, de tijdspanne waarin één en ander werd onderwezen is erg kort. Mijns inziens moet je direct nadat je de cursus hebt gevolgd deelnemen aan enkele Agile projecten.'

Kun je onderwerpen noemen die je in de cursus hebt geleerd en die je daarna hebt toegepast in je eigen praktijk?

Stephan: 'Helder krijgen van acceptatiecriteria en pas in een iteratie opnemen als voldaan is aan een Definition of Ready. Minder documenteren en vooral veel meer contact met klant en ontwikkelaars.'

Reinder: 'De cursus was intensief, doordat de nadruk lag op ervaren en gewoon doen. Je hebt ons goed begeleid, laten zwemmen maar je hebt ons ook geprikkeld.'

Jeroen: 'De stof behandelde ook de verschillende stadia van teamvorming. De persoonlijke feedback heb ik ter harte genomen. Ik ben mij nog meer bewust van de kracht van communicatie.'



→

Cursussen zijn prima om de basis van Agile testen te leren. De praktijk blijft de beste leermeester, maar gelukkig kunnen cursussen tegenwoordig steeds meer bieden om de Agile tester op de praktijk voor te bereiden. Dit is te danken aan enerzijds onderwijskundige ontwikkelingen op het gebied van diversiteit aan werkvormen die passen bij het doorlopen van verschillende leerstijlen en anderzijds aan nieuwe technologieën die integrale leertrajecten mogelijk maken, ook buiten het klaslokaal. Afgaande op ervaringen van onze cursisten is ons antwoord op de vraag 'Kun je Agile testen leren in een cursus?' een volmondig Ja! Daarna blijft het een belangrijke verantwoordelijkheid van het team, de Scrum Master (een tester wellicht?!) en eventueel de (e-)coach om aan dit leerproces een continu vervolg te geven in de eigen werkomgeving!

Bronnen:

- Bunt, M. van de (2012). *Certified Agile Tester*
- Opgehaald in augustus 2012 van: <http://www.trivento.nl/blogs/entry/certified-Agile-tester>
- Crispin, L. & Gregory, J. (2009). *Agile Testing: A Practical Guide for Testers and Agile Teams*. Addison-Wesley
- Improve Quality Services (2012). *Opleidingen*
- Opgehaald in augustus 2012 van: <http://www.improveqs.nl/opleidingen,3.html>
- Kolb, D. A. & Fry, R. (1975). Toward an applied theory of experiential learning. in C. Cooper (ed.) *Theories of Group Process*. London: John Wiley
- Lilienfeld, S., Ruscio, J. & Beyerstein, B. (2010). 'Busting Big Myths In Popular Psychology', *The Scientific American Mind*, March-April 2010
- Marselis, R. (2009). *Rik's Column: Agile*
- Opgehaald in augustus 2012 van: <http://www.TestNet.org/rik-s-column/Agile/menu-id-29.html>
- Tijman, A. & Jimmink, E. (2008). *Testen2.0, de praktijk van Agile testen*. Sdu uitgevers ←

EVENEMENTEN EN THEMA-AVONDEN

Najaarsevenement 2012

Dinsdag 2 oktober 09:00 – 21:00

Locatie: NBC, Blokhoeve 1, Nieuwegein

Hoe worden we nu echt Agile testers?

Dat is de rode draad van het TestNet najaarsevenement 2012.

Als keynotespreker opent Arie van Bennekum, één van de opstellers van het Agile manifest, het evenement met zijn visie op Agile ontwikkeling en testen.

's Avonds vertelt Peter Zimmerer, een testexpert uit Duitsland, over zijn ervaringen met Agile testen in de afgelopen 10 jaar (Peter was al in 2002 betrokken bij Agile testen)

's Ochtends zijn er maar liefst 5 tutorials/workshops (daarvoor kunnen leden zich apart aanmelden) en ook tijdens het evenement zijn er nog diverse tutorials en workshops op de aparte verdieping in de zalen 5, 6 en 7 onder de titel 'Agile Workshops'.

TestNet - Werkgroepen

Woensdag 14 november 2012 18:00 - 21:00

Locatie: NBC, Blokhoeve 1, Nieuwegein.

TestNet Noord 'Testomgevingen & ITIL'

Dinsdag 20 november 2012 18:00 - 21:00

Locatie Hajé Heerenveen, Schans 65, 8441AC Heerenveen.

TestNet - Interactie: TestGames

Maandag 26 november 2012 18:00 - 21:00

Locatie: NBC, Blokhoeve 1, Nieuwegein.

TestNet - thema-avond

Woensdag 12 december 2012 18:00 - 21:00

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