

Testen binnen een agile project is een teamsport: alle teamleden voeren test-activiteiten uit en ook de klant heeft hierbij een belangrijke rol. Wat betekent dit voor de tester zelf? Is hij daardoor overbodig geworden? Of heeft hij vooral een coördinerende rol? We belichten de verschillende rollen die een tester vanuit zijn specialisme kan vervullen in een agile project.

De rollen van de agile tester binnen het team

Onderlinge communicatie als sleutel tot succes

We baseren ons op onze eigen ervaringen als tester in diverse agile projecten. Wij merken dat er meer van ons gevraagd wordt dan alleen testanalyse en testuitvoer. Wij hebben in onze projecten naast onze rol als tester, onder meer de volgende petten opgehad:

- een rol als analist;
- een rol als teamspeler;
- een rol als kennisdeler;
- een rol als bruggenbouwer.

Ieder van deze rollen zullen wij nu behandelen.

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 nu vaak gemeengoed om een document als solist uit te werken.

Vanuit het oogpunt van volledigheid en juistheid van specificaties, is het veel makkelijker 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 gedachtekrinkel 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.

Een tester in een agile team is dus niet beperkt tot het reviewen van de requirements op volledigheid:

hij mag ook zelf meedenken, suggesties doen en risico's benoemen. Natuurlijk wordt er verwacht - net zoals dat ook voor de andere teamleden geldt - dat een tester zich snel het domein eigen maakt (om meerwaarde voor de business te kunnen genereren) of dat hij zich technisch verdiept (om meerwaarde voor het ICT-team te leveren). Die keuze hangt af van de competenties en interesse.

Tester in een agile project werken vaak met requirements die slechts op hoofdlijnen zijn uitgewerkt. Het is voor hem 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

Het aanvullen van requirements

Eric Jimmink: "Als tester in een agile team werk ik regelmatig user stories uit. Zo had ik bijvoorbeeld een aantal dataconversies binnen een project. Onze product owner kon op voorhand niet goed aangeven welke invulling "goed genoeg" zou zijn. Als team kozen we er voor om de implementatie pas drie weken later op te pakken, zodat in de tussentijd de details konden worden ingevuld. Toen heb ik de brondata geanaly-

seerd 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 c.q. uitwerken van



Anko Tijman

is Partner Agile Testen van Ordina. Het boek 'Testen2.0. De praktijk van agile testen' is zijn tweede boek over agile testen.



Eric Jimmink

is Testconsultant bij Ordina en co-auteur van het 'Testen2.0'-boek.

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 team committeert zich dan om iets te bouwen, waar de business richting aan heeft gegeven. 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 'Acceptance Test Driven Development' principe.

Het SMART (Specifiek, Meetbaar, Acceptabel, Resultaatgericht, Tijdgebonden) maken van de requirements kan worden gezien als het 'traditionele' reviewen: de tester laat zijn licht schijnen op de betekenis en interpretatie ervan. Oftewel: zijn ze niet voor meerderlei uitleg vatbaar en zijn ze concreet genoeg om op basis daarvan te gaan ontwikkelen. Het gaat dan puur om de functionele kant van de requirements: agile teams proberen technische beslissingen zo laat mogelijk te nemen, dus dat hoeft niet gereviewd te worden. 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.

Schermontwerp

Het uitwerken van het schermontwerp blijkt in de agile praktijk een bijzonder leerzame ervaring om samen met de product owner en eindgebruiker concreet toe te werken naar een werkend product. Door

de specificatie had ongetwijfeld geleid tot vertraging. Proactiviteit loont."

Anko Tijman: "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 gewens-

te functionele implementatie, zoals hoeveel producten en tot welk 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!"

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 acceptatietesten te concretiseren voorafgaande aan de implementatie. Door dit stelselmatig en uitvoerig te doen geven de testen een goede beschrijving van het systeem, die sturing geeft aan de gehele ontwikkeling. Zo wordt kwaliteit vooraf geborgd en is acceptatie nog slechts een formaliteit. Dit wordt ook wel 'Acceptance TDD' 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 houdt je de loop van feedback kort en neem je een bron van interpretatiefouten weg. Het team kan zo toewerken naar een oplossing die in één keer wordt geaccepteerd.

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.

Stand-up

Een belangrijke bron van informatie voor de teamspeeler is de stand-up meeting. Daar hoort de tester bijvoorbeeld welke (sub-)delen van functionaliteit testbaar zijn. Dan kan hij dus vroege feedback genereren door de

Veel en vaak testen helpt om fouten in het vervolgtraject te voorkomen.



Intensieve samenwerking zorgt voor begrip binnen het team voor de rol van de tester.

automatische of handmatige tests uit te voeren. Wanneer de tester dat doet kan hij 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 hij denkt dat hij daarin toegevoegde waarde kan leveren. Gelukkig leert de praktijk dat de denkwijze van de tester hierin veel waarde toevoegt.

Van de tester wordt dus verwacht dat hij 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 het team duidelijke werkafspraken te maken: soms heeft de tester rust en stabiliteit nodig en vaak moet hij snel kunnen reageren op veranderingen.

Taken overdragen

Naast de tester die als teamspeler gericht is op het team, gaat het bij de rol als teamspeler ook om de wederzijdse samenwerking. De tester moet bereid zijn soms testtaken over te dragen aan een teamlid dat 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 testen eenvoudig en inzichtelijk zijn opgezet, vergemakkelijkt dit het delegeren naar een collega die zelf geen testspecialist is. Eenvoud in de beschrijving van de testen 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 wordt verwacht;
- 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 de tester.

Op deze manier kan de tester er zeker van zijn dat zijn rol goed wordt uitgevoerd.

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 paarsgewijs 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. Een tester kan een ontwikkelaar meer leren over toe te passen testtechnieken, bijvoorbeeld op unittests.

Denk aan beslissingstabellen,

pseudocode of grenswaardenanalyse. Of hoe een goede handmatige blackbox test moet worden uitgevoerd, 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 zogenaamde Quick Reference Cards zijn beschikbaar op www.ordina.nl/agiletesten.

Paarsgewijs testen

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 de kennis van tester en materiedeskundige. De tester heeft een schat aan ervaring in het vinden van fouten in software. Vaak is een materiedeskundige of gebruiker er niet op gespitst en valideert men alleen wat wel werkt. Of men denkt voornamelijk vanuit de eigen ervaring en kennis ('error guessing'). De tester heeft een bredere blik vanuit de techniek en de kennis die hij in het ICT-project heeft opgedaan. Daarnaast kunnen testgevallen ter review worden aangeboden aan de deskundigen, zodat met de feedback die daaruit komt geborgd wordt dat de juiste tests worden uitgevoerd. Zodoende leert een tester meer over de business en is er 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

Zoek de activiteit met de op dat moment hoog

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 testen uitvoert en helpt

om requirements beter te maken, maar dat je bijvoorbeeld ook code gaat reviewen, documentatie schrijft of zelfs fouten oplost.

Eric Jimmink: "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 mis-



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 wordt bepaald welke systeemdelen met welke diepgang worden getest. Hoe hoger of belangrijker het risico,

des te dieper moet de test in verhouding zijn. Daarnaast helpt het nauwe contact om betere sturing op de acceptatietest te geven: je kunt beter vooraf voorstellen 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 juist dat er vroegtijdig getest wordt door de

De tester heeft in de praktijk vaak verschillende petten op.

klant. Die feedback is nodig om een project in de juiste richting te kunnen sturen. Wanneer de klant dus zijn rol als acceptant geen concrete handvatten weet te geven, ontbreekt er een stroom aan informatie in het project, terwijl dat juist wel nodig is om agile succesvol toe te passen.

Deze actieve rol geldt ook voor de toekomstige behorende partij; ook met hen zal de tester een nauwe werrelatie moeten onderhouden om te voorkomen dat er bij het naar productie gaan nog onverwacht 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 Scrumteams 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 Scrumteams. Sommige teams hebben uitstekende unittest suites, terwijl anderen ermee worstelen. Ook in dat soort situaties is de tester een toegevoegde waarde door de best practices met het team uit te wisselen. Bijvoorbeeld door een kennissessie te organiseren of gewoon een uurtje te plannen tussen ontwikkelaars uit de verschillende teams.

Communicatie

Deze vier verschillende petten van een agile tester hebben een belangrijk ding gemeen: het gaat uitdrukkelijk om communicatieve vaardigheden. Dit komt dus bovenop de 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. «

De tester draait volledig mee in het team en laat anderen ook beter hun werk doen.

gste toevoegde waarde

schien 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 te zoeken op Internet vond ik de tip die leidde tot het verhelpen van het probleem.”

Anko Tijman: “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 te veel 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.”

Vervolg

In een volgend artikel over agile testen zullen we een blik in de toekomst werpen over wat organisaties kunnen verwachten aan rendementen wanneer ze agile inclusief het testen hebben ingericht en de agile aanpak de standaard is geworden.