

Agile softwareontwikkeling is dynamisch, kortcyclisch en interactief. Het testen van software is veelal gebaseerd op bevroren documenten en wordt uitgevoerd op uitontwikkelde software door een apart testteam. Toch kunnen deze twee disciplines uitstekend met elkaar overweg in een nieuwe testaanpak: agile testen. Daarin gaan dynamiek, samenwerking en kwaliteit hand in hand.

Tijd voor een nieuwe manier van testen

Agile werken en testen gaan prima samen

Dit is het eerste in een reeks van artikelen over 'agile testen'. Anko Tijman en Eric Jimmink, auteurs van het boek 'Testen2.0 – de praktijk van agile testen' beschrijven hierin hun ervaringen en inzichten rondom agile testen. Zij vinden agile projecten zeer kwaliteitsbewust en dat zullen zij in dit artikel onderbouwen. Toch kan een agile team niet zonder een gespecialiseerde tester. De basis van agile testen wordt uitgelegd aan de hand van een viertal basiswaarden.

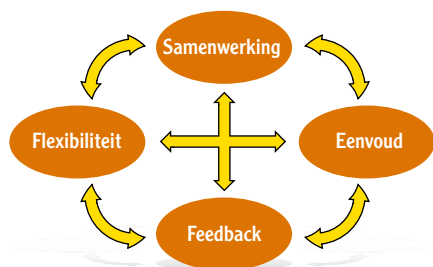
het project en requirements aan te passen aan de actuele wensen van de business. Door de klant een actieve rol in het project te geven kan er steeds gevalideerd worden op de (misschien wel wijzigende) bedrijfs- en systeemdoelstellingen. Wij merken in onze dagelijkse praktijk dat deze aanpak daadwerkelijk in staat is om binnen tijd en binnen budget software van goede kwaliteit op te leveren.

Testen is inzicht geven in de kwaliteit en voortgang van het product door middel van verificatie ('het product goed bouwen') en validatie ('het goede product bouwen'). Testen is de check in de Deming cirkel Plan-Do-Check-Act. Testen is niet alleen een activiteit maar ook een professie. Naast het voorbereiden en uitvoeren van tests, kennen we de zogenaamde teststrategie (hoe diepgaand test je de verschillende systeemdelen), testtechnieken (welke testgevallen heb je nodig om de gewenste diepgang te halen) en testplannen (hoe zorg je ervoor dat het testen efficiënt gebeurt).

Om zeker te weten dat je goed bezig bent met het ontwikkelen van nieuwe functionaliteit of aanpassingen, zul je vaak controles moeten uitoefenen: veel en vaak testen dus! Wij zien dit in de agile context terug; testen is een belangrijk onderdeel in het agile proces en het team is 'test infected'. Dat zullen we toelichten.

Agile teams zijn 'test infected'

Wij zien in onze agile projecten heel veel aspecten van testen en kwaliteitszorg terugkomen waar wij als testers warm voor lopen. Eén van de belangrijkste technische aspecten is het gebruik van unit tests. De juiste werking van de code wordt geborgd



De vier essentiële waarden in een Testen 2.0 proces.

Agile en testen

Dit is zeker niet het eerste artikel in dit blad over agile softwareontwikkeling, maar het lijkt ons goed om te benoemen wat wij daaronder verstaan. Agile softwareontwikkeling is een multidisciplinaire, iteratieve projectaanpak waarbij er in korte cycli werkende, waardevolle software wordt opgeleverd. Het team werkt in een heel kort tijdsbestek toe naar werkende eindproducten die frequent worden teruggekoppeld aan de opdrachtgever.

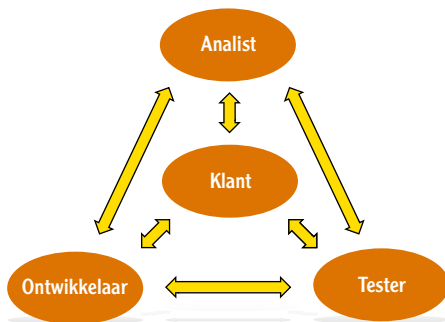
Aan het einde van een iteratie selecteert de klant de requirements voor de volgende iteratie. Dit stelt de klant in staat om tussentijds sturing te geven aan



Anko Tijman
is Partner bij Ordina.
bij Ordina



Anko Tijman
is Partner bij Ordina.
bij Ordina



De vier elementaire rollen binnen Testen 2.0.

door uitgebreide unit tests, dat zijn testgevallen op code

niveau. De onderhoudbaarheid van de code wordt bevorderd door gezamenlijk de code te beheren, eenvoud na te streven, elkaars code te reviewen of die zelfs in tweetallen te schrijven (het zogenaamde pair programming). Daarbij worden standaarden qua vorm en inhoud gehanteerd. Een agile ontwikkelaar is niet alleen trots op de gemaakte code, maar ook op de unit tests die daarbij horen. Een manier om ervoor te zorgen dat er geen code voorkomt zonder een bijbehorende unit test, is de afspraak om die testen eerst te schrijven en daarna pas de code! Dat is de kern van Test Driven Development.

Een ander belangrijk technisch aspect in agile is continuous integration: het in kleine stapjes voortdurend bouwen van een compleet geïntegreerde applicatie. Na de integratie wordt vaak meteen de regressietest geautomatiseerd uitgevoerd. Deze

Het vangnet

In een agile project worden veel tests door ontwikkelaars en testers geautomatiseerd, zodat regressiefouten direct worden gedetecteerd. Het team heeft als het ware een geautomatiseerd vangnet om onbedoelde wijzigingen tegen te gaan. Dit geeft vrijheid om code te refactoren: om de code op te schonen en bijvoorbeeld simpeler te maken, nadat het systeem lijkt te werken. In het verleden (jaren '80, begin jaren '90) was er een Gouden Regel: "If it works, don't fix it!". Dat principe is dus volkomen passé. Het is juist heel goed om de code op te schonen, juist om deze naar de toekomst toe onderhoudbaar te maken. En in een agile omgeving verandert er veel. De requirements zullen tijdens het project regelmatig worden herzien, wat veelal ook wijzigingen in de reeds afgeronde code zal inhouden. De extra inspanning die het team investeert in het geautomatiseerd onderhoudbaar maken en houden van de code is daarmee al snel terugverdiend. Voor de goede orde: handmatige (aanvullende) regressietesten zijn meestal nog steeds nodig. Het is dus een vereiste dat wijzigingen in een team worden gecommuniceerd, zodat de tester op de hoogte is en waar nodig 'getriggerd' wordt om aanvullende testen uit te voeren.

techniek creëert korte feedback loops; het team weet exact hoe het met de ontwikkeling van de applicatie staat. Eventuele fouten moeten direct hersteld worden, want werkende software is de norm in een agile project. Dat is de enige juiste manier van het meten van voortgang! Met deze techniek moeten testers overigens opletten: korte feedback loops zijn waardevol en het is goed om exact te weten waar het team staat. Maar pas op dat er niet zo vaak geïntegreerd wordt, dat je daardoor geen fatsoenlijke test meer kunt uitvoeren op een versie. De exacte balans tussen integreren en testen verschilt per project en per team.

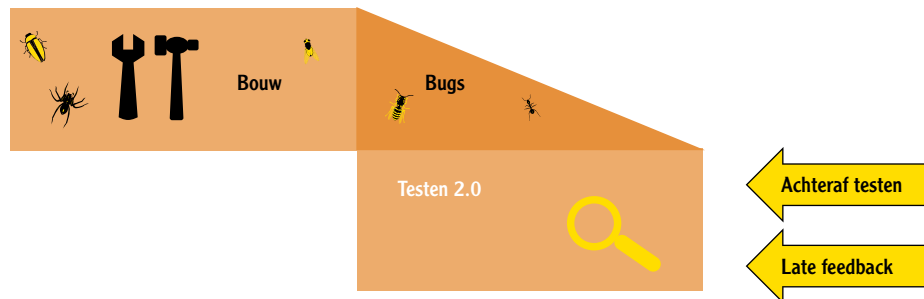
Testvirus ook bij de klant

Unit tests en continuous integration zijn (slechts!) twee aspecten van de wijze waarop de ontwikkelaars in een agile project met kwaliteit en testen omgaan. Maar ook bij de klant is er sprake van 'geïnfecteerd zijn met het testvirus'. Doordat de opdrachtgever actief meeloopt met het project, kan hij op een informele manier al heel veel over het systeem te weten komen. Daarnaast voeren materiedeskundigen al tijdens de iteraties acceptatietests uit. Voor opdrachtgever en opdrachtnemer houdt het tussentijds testen in dat er zeer frequent feedback wordt gegeven, zodat het systeem steeds meer daadwerkelijke waarde voor de business heeft. Zodoende kan het uiteindelijke product in één keer goed worden opgeleverd. Ook doet het team zijn uiterste best om iedere iteratie te voldoen aan de uiteindelijke acceptatiecriteria. Het is aan de opdrachtgever de taak om deze op te stellen, maar het team mag daar uiteraard bij helpen. Waar het uiteindelijk om gaat is dat de acceptatiecriteria iedere iteratie gehaald worden. Dit om exact te weten waar je staat met de ontwikkeling van het systeem. Je wilt immers niet in de laatste fase van het project geconfronteerd worden met werk dat al veel eerder gedaan had moeten worden (zoals vaak gebeurt met het voldoen aan de acceptatiecriteria). Elke iteratie wordt afgesloten met een demo van de resultaten aan alle stakeholders. Hierbij kunnen ook eindgebruikers het product zien groeien en daarmee wordt draagvlak gecreëerd voor de acceptatie van het product als geheel.

Geen fase achteraf

Kijken we specifiek naar de rol van de tester dan zien wij in onze praktijk dat een heleboel zaken vanzelfsprekend zijn voor het agile ontwikkelteam. Zo worden er vanaf dag één op het project al tests geschreven en uitgevoerd. Dat betekent dat testen niet langer een fase achteraf is, maar een integraal onderdeel van het ontwikkelproces is. Daarmee is het testen dus meteen ook van het kritieke pad af. Omdat er veel en vaak getest wordt, heeft het team veel meer aandacht voor de specifieke testproblematiek zoals de geschiktheid van de testomgeving

Testen is niet langer een fase achteraf, maar een integraal onderdeel van het proces.



Het herstellen van fouten uit de systeemtest in een traditioneel ontwikkelproces.



Binnen Testen 2.0 wordt er parallel getest, waardoor er sneller bugs worden gevonden en minder gevolgfouten worden gemaakt.

en de volledigheid van de testdata. Wij hoeven als testers niet langer meer te 'zeuren' om aandacht: de focus van het team ligt er al. Daarnaast is de traditionele valkuil van het testen, de gebrekkige testbasis, ook op afdoende wijze ondervangen. Namelijk: door de actieve rol van de klant in het proces. De testbasis bestaat niet langer uit documentatie maar ook uit communicatie met de klantvertegenwoordigers. Aan de testers is het dan de taak om die kennis te verzamelen. En mogelijk zelfs te bundelen in het testontwerp. Dat er vooraf niet altijd een gedetailleerd functioneel ontwerp is, is dan niet meer zo van belang. Het team kijkt vanuit

De testdialoog met de klant

De klantvertegenwoordiger (een materiedeskundige, functioneel beheerder of een eindgebruiker) heeft een belangrijke controlerende en sturende taak in de vorm van het opstellen en uitvoeren van acceptatietesten. Hij heeft immers het beste inzicht in de daadwerkelijke waarde van het systeem voor 'de business'. De tester uit het team kan deze acceptatietesten faciliteren, en bijvoorbeeld helpen bij het automatiseren van de tests. De tester zoekt de dialoog met de klant op om het bedrijfsbelang achter de requirements helder te krijgen en hierop in te spelen tijdens het testen. Die informatie heb je nodig om op risico's gebaseerd te kunnen testen en dus met beleid de testinspanning te kunnen differentiëren. Het opstellen van de acceptatiecriteria is in principe voorbehouden aan de klant. Die formulering zal zelden helemaal duidelijk en volledig zijn. De tester kan het team helpen om samen met de klant de criteria eenduidig te maken. Het doel is immers dat het team exact weet hoe het juiste product er uit moet zien.

de verschillende disciplines (ontwerp, bouw, test en eindgebruikers) naar het eindresultaat. Aannames worden daarbij onderling geverifieerd. Dat voorkomt dure herstelwerkzaamheden. Dat betekent dat testers (eindelijk!) invloed hebben op het ontwerp en de code. Precies zoals we dat altijd al hadden willen hebben. Wij voelen ons als testers in agile teams zeer gewaardeerd – en dat is een goed gevoel!

En de tester...?

Kun je in een dergelijk 'test infected' team nog wel een rol spelen als tester? Uiteraard vinden wij van wel! Want het feit dat er veel getest wordt wil natuurlijk niet zeggen dat er effectief getest wordt.

Ontwikkelaars stoppen hun creatieve energie in het bouwen van het product en beogen met de testen die zij uitvoeren aan te tonen dat het door hun gebouwde systeem voldoet aan de specificaties. Een tester zal juist op creatieve wijze proberen om het tegendeel aan te tonen: hij gaat 'op zoek naar de fouten'. Een klantvertegenwoordiger zit qua attitude doorgaans tussen die twee extremen in en test op een andere manier; hij of zij heeft essentiële bagage in de vorm van materiekennis en test vanuit het bedrijfsproces. Deze drie invalshoeken vullen elkaar aan. Je hebt ze allemaal nodig om effectief te testen. Daarom moet je multidisciplinair werken – en dus ook multidisciplinair testen.

De rol van de tester is dus breed: hij kan zowel met ontwikkelaars op het technische vlak als met eindgebruikers op het functionele vlak samenwerken. Daarnaast kan hij ook nog eens een vakinhoudelijke verdieping aanbrengen in het testen. Zo kan de tester unit tests reviewen en integratie- en regres-

Testen is feedback en daarmee de voornaamste toegevoegde waarde van het testen.

sietests opbouwen. Eenvoudige testtechnieken als equivalentieklassen of Exploratory testen zijn nog maar weinig bekend bij ontwikkelaars. Over de invulling van de rol van de tester in een agile project kan veel gezegd worden. Daaraan gaan we in een van de volgende artikelen dan ook specifiek aandacht aan besteden.

De vier kernwaarden

Wat is de essentie van agile testen? Als we de werkwijze van het gehele agile team en in het bijzonder de agile tester willen samenvatten in enkele begrippen, dan komen wij op de volgende: Samenwerken, Feedback, Eenvoud, Flexibiliteit.

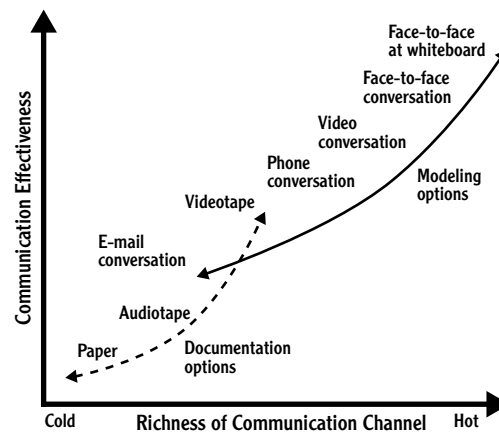
Dit zijn vier waarden die in een agile testproces essentieel zijn. Agile testen is geen activiteit die solistisch kan worden ingevuld. Het testen is niet langer het einde van de pijplijn maar vindt parallel plaats, Samenwerken is dus essentieel. Collega's van andere disciplines zitten in hetzelfde team, in dezelfde kamer, met hetzelfde doel voor ogen. In agile testen is er dus geen sprake van een aparte testafdeling, want afscheiding creëert blokkades in communicatie. Iedere discipline mag zichzelf beschouwen als het middelpunt van de onderlinge samenwerking en dat geldt dus ook voor de testers.

Testen is feedback, het is de voornaamste toegevoegde waarde van het testen. Alle kleine stapjes die in het project gezet worden, moeten zo vaak en zo veel mogelijk gevalideerd worden. Dan weet je namelijk exact waar je staat als team. Waarmee levert het testen de meeste feedback? Inderdaad, met het daadwerkelijk uitvoeren van testen – zeer tastbare resultaten! Ook het draaien van tests is werkende software en dat is de heilige graal van voortgangsmeting in een agile project.

Al dat samenwerken en al die feedback resulteert in een grote behoefte aan flexibiliteit. Door snel te schakelen kan die feedback gegeven worden waar het team op dat moment het meeste behoefte aan heeft. Dit stelt forse eisen aan de manier waarop het werk georganiseerd is: het moet zodanig ingericht zijn dat er snel prioriteiten gesteld kunnen worden. Je moet immers snel beslissingen nemen in een hoog interactieve omgeving. Dit moet je vervolgens ook weer niet op het moment van de beslisser zelf af laten hangen, maar het werk inrichten op basis van prioriteiten.

Tenslotte is eenvoud heel belangrijk. Een agile team streeft naar eenvoud omdat complexiteit zo'n slecht alternatief is. Dat communiceert moeilijker, dat maakt het programmeren en testen zo moeilijk – om over onderhoud nog maar niet te praten! Daarnaast worden in agile teams basistaken soms aan elkaar overgedragen, ook dan geldt dat eenvoud regeert. Hoe sneller je immers iets oppakt, hoe sneller het uitgevoerd kan worden. En uiteraard geldt dit ook voor het testen: testers zullen dus moeten leren (als een extra kwaliteitsattribuut) om hun eigen werk ook door het oog van de ander te bekijken. En wees niet bang dat een ander er moet jouw kennis van doorgaat, want dat anderen meer leren over het vak testen is toch alleen maar mooi?

Het moge duidelijk zijn: wij vermaken ons prima als testers in agile projecten. Ondanks dat ze op het oog niet heel veel gemeen hebben. Schijn bedriegt, maar als je er doorheen weet te prikken dan ga je steeds meer overeenkomsten zien. In onze volgende artikelen gaan we dieper op agile testen in. Dan behandelen we onder andere de dagelijkse werkzaamheden van de agile tester, de kernproducten van agile testen en het profiel van de gespecialiseerde agile tester. «



De effectiviteit van informatiekanaalen volgens Alistair Cockburn.