

De vier waarden en twaalf principes van het agile manifesto staan neutraal tegenover de testaanpak. En ook de op de manifesto gebaseerde agile softwareontwikkelomgevingen geven nauwelijks handvatten voor het testen. Het zal dan ook geen verbazing wekken, dat het invoeren van een testproces in een agile softwareontwikkelomgeving niet altijd soepel verloopt.

Adaptiviteit is sleutel tot het Agile-testen

Testsoorten integreren in ontwikkelproces

Testen is niet alleen een uitermate belangrijke activiteit, maar om effectief te zijn moet het ook nog eens volledig geïntegreerd worden in de gekozen agile ontwikkelmethode. Het agile manifesto zelf geeft nauwelijks handvatten hoe testen met een agile softwareontwikkelomgeving geïntegreerd kan worden. In de praktijk blijken in agile omgevingen dan ook vaak de volgende testuitdagingen te bestaan:

1. Wat met testen te doen als de agile ontwikkelmethode niet volledig is geïmplementeerd?
2. Hoe kan het testproces met een agile softwareontwikkelomgeving worden geïntegreerd?
3. Hoe kan de juiste (test)balans m.b.t. de waarden van het manifesto worden gevonden?
4. Hoe past het gebruik van een gestructureerd testproces in een agile omgeving?

In de beperkte ruimte van dit artikel geven wij in globale bewoordingen oplossingsrichtingen aan die in uw specifieke agile softwareontwikkelomgeving kunnen worden toegepast.

Manifesto als uitgangspunt

Het gaat vaak al mis als een organisatie de gekozen agile ontwikkelmethode niet volledig implementeert. Ze kiezen slechts enkele aansprekende waarden en principes en starten daarmee de implementatie van een agile aanpak. Het manifesto is echter geen checklist waar je willekeurig iets van kan kiezen. Hoewel de organisatie vrij is in de wijze van implementeren (vandaar ook het scala aan agile aanpakken), moeten uiteindelijk wel alle waarden geïmplementeerd worden. Wat kunt u hieraan doen? Begin met het simpelweg lezen van een boek of het aanstellen van een agile-expert als coach. Deze coach kan, totdat

iedereen gewend is aan de nieuwe aanpak, training en ondersteuning geven en tevens zorgen dat iedereen de waarden op dezelfde wijze interpreteert.

Onvolledig implementeren van een agile ontwikkelmethode is op zich geen testuitdaging, maar heeft wel gevolgen voor het testen. Ook alle voorgestelde testactiviteiten moeten passen binnen de waarden en principes van het manifesto.

Het integreren van het testproces met een agile softwareontwikkelomgeving bezorgt sommige organisaties nogal wat hoofdbrekens. Wij lichten er twee uitdagingen uit:

- **Hoe kan testen worden geïntegreerd met het softwareontwikkelproces?**

In tegenstelling tot lineaire methoden, waar testsoorten zoals systeemtest en acceptatietest vaak fasen ná oplevering van de software zijn, worden bij een agile aanpak deze testsoorten in het ontwikkelproces geïntegreerd. Dit betekent dat testen geen aparte fase meer is, maar dat testen een continue activiteit van de agile teams is. Hiervoor is een specifieke implementatie van testsoorten nodig.

Bijvoorbeeld doordat in iedere iteratie de projectleden vanuit hun eigen perspectief de technische-, functionele- en bruikbaarheidstests uitvoeren. Deze tests worden én vanaf het begin van de iteratie én parallel aan elkaar uitgevoerd. In feite zijn dit gecombineerde unit-, systeem- en acceptatietests. Door in de iteraties 'continuous integration' toe te passen, betekent dit dat na de laatste iteratie alle tests zijn afgerond. Immers in de laatste iteratie zijn de nieuwe features getest en is er een regressietest op de reeds bestaande features uitgevoerd. Soms worden



Leo van der Aalst en
Rik Marselis

na oplevering van de release nog andere tests uitgevoerd. Een ketentest bijvoorbeeld.

Het gebruik van testtools is in een agile omgeving onmisbaar om de gewenste performance van de agile teams te realiseren. Immers, in de vaak korte iteraties moeten toch alle testactiviteiten uitgevoerd worden. Daarom worden zeker de regressietests bijna altijd geautomatiseerd uitgevoerd.

• Hoe kan testen worden geïntegreerd met de projectteamactiviteiten?

In een agile ontwikkelomgeving geeft de manager het multidisciplinaire team het vertrouwen dat ze zelf het project tot een goed einde brengen. Het goed en tijdig opleveren van de software is een teamverantwoordelijkheid. En hoewel een professioneel tester met testexpertise in het team aanwezig moet zijn, wil dit niet zeggen dat alle testactiviteiten door deze tester moeten worden uitgevoerd. De tester is niet meer de traditionele ‘quality gatekeeper’, maar werkt met alle teamleden samen om continu informatie te kunnen geven over de voortgang en het voldoen aan de businesswensen van het product. Door de informatie die testen oplevert is deze activiteit nu eigenlijk de stuwende kracht van het project.

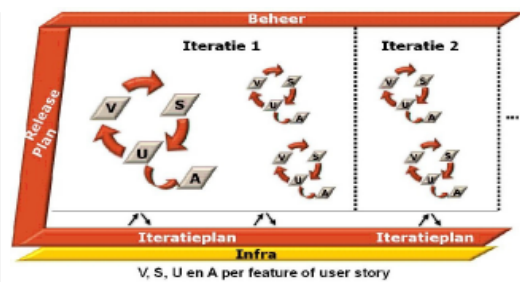
Alle teamleden moeten bereid zijn testactiviteiten uit te voeren (net zoals dat de tester soms bij ander werk bijspringt) om vóór het einde van de iteratie het afgesproken werk gedaan te krijgen. Hierbij helpt het om in de ‘definition of done’ (dit zijn de exitcriteria) ook de testaspecten op te nemen.

Kies bewust

Het vinden van de juiste (test)balans met betrekking tot de vier waarden van het manifesto is vaak lastig. Het team moet bijvoorbeeld een balans vinden van wat wel en wat niet gedocumenteerd moet worden en op welke manier de beperktheid van de documentatie kan worden gecompenseerd. Het vinden de juiste balans heeft alles te maken met ‘common sense’ en met het maken van bewuste keuzes.

Werkende software wordt belangrijker gevonden dan uitgebreide documentatie, dan kan het ‘gemis’ aan documentatie worden opgevangen door meer te communiceren. Aan de andere kant, als documentatie belangrijk is voor een controlerende instantie en gemaakt moet worden, dan moet dit ‘gewoon’ in de iteratieplanning worden opgenomen én gemaakt. Ook al lijkt dit strijdig met één van de principes, maar het is juist een aspect van agile werken, om adaptief om te gaan met de omstandigheden.

Het testproces moet op de agile omgeving worden toegesneden, rekening houdend met de specifieke eigenschappen van agile ontwikkelen. Adaptiviteit is één van de vier TMap NEXT essenties. Hierdoor is het heel goed mogelijk om zowel het faseringsmodel



als de testactiviteiten aan te passen aan de agile ontwikkelomgeving. Voor een aangepast faseringsmodel zie figuur 1.

Enkele voorbeelden van ‘geagileerde’ testactiviteiten zijn:

- Maak bij de start van het project, in samenhang met het releaseplan, een compact releasetestplan met daarin de iteraties en een globale teststrategie. Dit plan wijzigt gedurende het project op basis van de voortschrijdende inzichten. Bij het begin van elke iteratie maakt het team een iteratietestplan. Dit kan heel basaal door op een whiteboard een tabel te zetten, waarin gedurende de iteratie steeds meer kolommen worden ingevuld. Denk bijvoorbeeld aan het bepalen van een risicoklasse per combinatie van kwaliteitsattribuut en user story. Geef in deze tabel aan door wie, hoe zwaar en met welke testontwerptechniek het risico wordt afgedekt.
- Continu neemt de tester (of een teamlid met een testrol) een proactieve houding aan bij het beoordelen of de testbasis (requirements, user stories, enz.) consistent en compleet is, dus geschikt als basis om testgevallen te schrijven. Direct vanaf de start van het project moet de tester aan, onder anderen, de architect, programmeur en gebruiker vragen stellen over de user stories of features. “Wat gebeurt er in die situatie en wat in zo’n situatie?” Als de tester de antwoorden bevredigend vindt, is daarmee de intake testbasis afgerond. In de agile situatie hoeft geen rapport detailintake te worden geschreven.

De tester verwerkt de antwoorden ook in de testgevallen, hoewel ze niet altijd in de systeemdokumentatie zijn opgenomen. Hierdoor gebeurt het nog wel eens dat in een agileproject meer informatie van het systeem vastligt in de testgevallen dan in de systeemdokumentatie. Het mag duidelijk zijn dat in deze situatie conservering van de testware speciale aandacht verdient. Immers het ‘weggooien’ van de testware betekent tijd-, kapitaal- en kennisvernietiging.

De sleutel

Niet het ‘weggooien’ van bestaande testaanpakken, maar het adaptief omgaan met in de testpraktijk bewezen principes, processen en activiteiten, dat is de sleutel tot het succesvol integreren van het testen in een agile softwareontwikkelomgeving. «

Figuur 1. Agile en het TMap NEXT faseringsmodel.

Leo van der Aalst is research & development manager bij Sogeti en lector Software Quality and Testing aan de Fontys Hogeschool. Hij is te bereiken via leo.vander.aalst@sogeti.nl.

Rik Marselis is senior testconsultant bij Sogeti en bestuurslid van de vereniging TestNet en de stichting BNTQB. Hij is te bereiken via [rik.marselis@sogeti.nl](mailto:marselis@sogeti.nl).