

Quick Reference Card

FAQ over Agile Testen

Doel van de QRC:	Antwoord en inzicht geven in de meest gestelde vragen rondom agile testen. Dit op basis van gesprekken met beginnende agile testers en het topic op www.testforum.nl
Scope:	We richten ons in deze QRC specifiek op agile testen , niet op algemene agile vragen.
Opgesteld door:	Werkgroep Agile Testen van TestNet
Feedback:	agiletesten@testnet.org

Vragen en antwoorden

Vraag	Antwoord
<i>Heb je bij agile testen een bevindingendatabase nodig?</i>	Dat hangt er van af. Sommige –vaak kleinere- teams hebben genoeg aan post-its op het taskboard. Vaak is een eenvoudige tool voldoende. Stelregel is: voorkom dat gevonden bugs vergeten kunnen worden en denk na over welke behoefte het team aan metriecken heeft.
<i>Hoe weet je nu of datgene dat wordt opgeleverd 'kwalitatief' goed genoeg is?</i>	Een agile team gebruikt hiervoor de Definition of Done: het gezamenlijk overeengekomen kwaliteitskader. Iedere feature is pas 'done' als het aan die criteria voldoet. Zie de QRC Definition of Done.
<i>Hoe kan een agile tester developers assisteren bij het opzetten van unit-tests?</i>	Minimaal twee manieren: <ul style="list-style-type: none"> - je helpt door testtechnieken te gebruiken om de werking van de functionaliteit helder te krijgen. Denk aan het gebruik van een beslissingstabel bij complexe beslissingen. Spreek jouw testsituaties door met de ontwikkelaar; - Review de unit tests van de ontwikkelaars op dekkingsgraad a.d.h.v. testtechnieken en bespreek de resultaten.
<i>Hoe kun je tests automatiseren terwijl je GUI nog volop in ontwikkeling is?</i>	In agile teams worden veelal testtools gebruikt die net onder de GUI testen, zoals Fitnesse. Daarmee ben je dus niet langer afhankelijk van de GUI. Sowieso wordt er veel en vaker geautomatiseerd getest in agile omgevingen, omdat je tests veel vaker moet herhalen (door de iteratieve werkwijze).
<i>Als je in een agile team, zeer nauw met elkaar samenwerkt, ga je dan niet al snel in dezelfde valkuilen stappen als ontwerpers en ontwikkelaars?</i>	Natuurlijk bestaat dat risico, maar als tester moet je altijd kritisch kunnen blijven denken. Daarvoor hoeft je niet in een apart team te zitten. Daarbij is de waarde van jouw testkennis veel groter, dan je waarde als onafhankelijk beoordelaar. Leer jouw teamgenoten dus testtechnieken en de waarde van testen in hun werk. Ze doen hun werk beter als ze meteen doen wat jij als tester ook doet.
<i>Hoe zit het met de verhouding Ontwikkelaar: Tester binnen een redelijk goed lopende agile team?</i>	Houd allereerst dezelfde verhouding aan als nu in je teams. Je leert snel in agile teams, dus ook of je te veel of te weinig testers hebt. Verder is de ratio erg afhankelijk van: <ul style="list-style-type: none"> - Het soort project/ klant - De omvang van het totale projectteam - De vaardigheden van de teamleden
<i>Kan je ook agile testen in een waterval project?</i>	Ja en nee. Je kunt zeker practices van agile testen toepassen in traditionele projecten, zoals AcceptanceTDD, unit testen reviewen, korte feedback loops vanuit de systeemtest en continue acceptatie door de klant faciliteren. Aan de andere kant wordt je project pas echt effectief als iedereen met de neuzen dezelfde kant op staat en de agile waarden (interacties,

Quick Reference Card

	samenwerken, werkende software en kunnen reageren op veranderingen) onderschrijft. Agile testen in een watervalproject zal daarmee altijd suboptimaal zijn, maar soms ook beter dan puur traditioneel testen.
<i>Hoe kan ik agile vanuit testperspectief het best promoten in een nog traditioneel denkende organisatie?</i>	Testen is risicomanagement, en agile testen is het creëren van korte feedback loops vanuit testen. Als tester constateer je tekortkomingen in requirements – waarom zou je niet samenwerken met een analist? Idem dito met het ontwerp, review het niet achteraf maar creëer eens een kans om er samen met de ontwerper aan te werken. En waarom zou jij unit tests niet kunnen reviewen? Ook de systeemtest zou je in kleine iteraties kunnen opdelen, zodat je voorkomt dat het integreren van de code een lijdensweg wordt.
<i>Zijn agile en Exploratory testen hetzelfde?</i>	Nee. Agile testen is een testaanpak voor software ontwikkeling. Exploratory testen is een testtechniek waarbij de tester zijn testgevallen ontwerpt tijdens testuitvoering en hierbij verkregen resultaten gebruikt worden om nieuwe, nog betere testgevallen te ontwerpen.
<i>Wanneer kun je beter wel voor een agile testen aanpak kiezen?</i>	In ieder agile project is agile testen de voorkeursaanpak. Een traditionele testaanpak werkt daar contraproductief omdat het testen pas laat feedback en relevante testkennis oplevert voor de overige disciplines. In een traditioneel project kun je elementen van agile testen toepassen als het team daarvoor open staat.
<i>Wanneer kun je beter niet voor een agile testen aanpak kiezen?</i>	Als er veel weerstand in de organisatie is ten aanzien van transparantie, flexibiliteit en samenwerken met betrekking tot testen. Ieder team dat zich agile noemt, moet de agile testen aanpak faciliteren.
<i>Als documentatie minder belangrijk wordt, en ik minder documentatie oplever, hoe laat ik dan mijn bijdrage zien?</i>	Door meewerkend teamlid te zijn dat waardevolle informatie verschaft met betrekking tot volledigheid en duidelijkheid van requirements, korte feedback loops genereert door op kleine stukjes software al tests uit te voeren en door de klant proactief bij het acceptatietesten te betrekken. Richt je op de informatie die mensen delen in plaats van uitsluitend documentatie.
<i>Als ontwikkelen sneller gaat dan testen, hoe krijg je de developers dan zo ver om te helpen met testen?</i>	Er is een gezamenlijke verantwoordelijkheid voor het opleveren van kwaliteit, de Definition of Done. Daar hebben dus ook de ontwikkelaars hun commitment aan gegeven. Spreek het team hierop aan.