

De noodzaak voor een agile teststrategie

Agile teams zijn zelf verantwoordelijk voor het leveren van kwaliteit. Desondanks is het aan te bevelen om een overkoepelende teststrategie te hebben, betoogt Derk-Jan de Groot.

Derk-Jan de Groot



'Kijk', legt de projectmanager uit, 'dit project is erg belangrijk voor de organisatie. De geplande aanpassingen doorsnijden alle systemen, maar ze zijn noodzakelijk om ook in de toekomst onderscheidend te zijn. Ik wil erop kunnen vertrouwen dat na elke release de *value streams* nog steeds werken, maar hoe weet ik of dit zo is? De meeste van onze teams zijn agile, maar niet allemaal. Als ik begin over een teststrategie, dan zeggen ze dat dit oldschool en achterhaald is.' Hij kijkt mij vragend aan. Ik ben uitgenodigd om mee te denken over de testaanpak en nu wordt er een antwoord verwacht.

Met de komst van Agile verandert er veel binnen de softwareontwikkeling. De aanpak brengt nieuwe uitdagingen met zich mee voor de testers. De grootste verandering is misschien wel dat de agile teams zelf verantwoordelijk zijn voor het leveren van kwaliteit. Michael Sowers verwoordt het heel goed in zijn blog van afgelopen september: *'In Agile, the accountability for the right level of quality delivered at the right time belongs to the collective team. The team should embrace the best skill sets of each contributor and plan for quality and testing at every step of the release and within each sprint.'*

Gevolg hiervan is dat de verantwoordelijkheid van het testen niet enkel meer ligt bij de testers maar bij het hele team. Daarbij willen we zo veel mogelijk van de tests onderbrengen in de sprint. De noodzaak om ze te organiseren als aparte deelprojecten valt weg en testmanagers zijn niet meer nodig.

Landkaart inkleuren

Een voor de hand liggende reactie op de vraag van de projectmanager zou dan ook zijn om uit te leggen dat we helemaal geen overkoepelende strategie nodig hebben en dat de teams zelf maar moeten nadenken over hoe ze hun tests aanpakken. Geen strategie hanteren heeft het voordeel dat we de overhead en het aantal artefacten sterk reduceren. We zijn verlost van de *big design up front* en kunnen de tests *just in time* definiëren op basis van de kennis die we op dat moment hebben.

Ondanks deze voordelen, geloof ik er niet zo in. Ik heb een groot aantal ontwikkelteams gesproken die kwaliteit echt wel serieus nemen. Maar als ik aan het begin van een sprint vroeg hoe ze het testen gingen aanpakken, waren de reacties vaak in de trant van 'we kennen onze applicatie en weten heus wel wat we doen', 'we doen dit al jaren' en 'vertrouw ons nou maar'. Een helder antwoord kreeg ik zelden.

Testen is meer dan fouten vinden. Tests dienen ook aan te tonen wat er wel werkt. Ze geven niet alleen inzicht in de productrisico's maar ook vertrouwen. De risico's en het vertrouwen dragen bij aan de acceptatie van de oplossing. Daarnaast levert testen inzicht in de voortgang van de realisatie. Dit stelt de organisatie in staat om bij te sturen, mitigerende acties te nemen of werkitems te herprioriteren.

De business wil de gevraagde oplossing – een release, een *minimal viable product* of een projectresultaat – met vertrouwen accepteren. Daartoe zullen we moeten weten welk soort tests we allemaal moeten uitvoeren.

Dit kunnen we definiëren in wat ik percelen noem. Deze beschrijven welke soorten tests we moeten uitvoeren én welke informatie de uitvoering ervan oplevert, maar niet welke tests precies en op welke manier; dat is aan de agile teams. Gedurende de realisatie kleuren we de landkaart perceel voor perceel in. Dit doen we op basis van de testresultaten die we verzamelen bij de teams. Zo ontstaat er gaandeweg vertrouwen in de oplossing.

Teams die op eigen houtje testen en zich met een 'dit doen we altijd zo' onthouden van hun verantwoordelijkheid om inzicht te geven in de tests die ze plannen en uitvoeren, ondermijnen dit proces. Zij moeten leren dat alleen de juiste tests doen niet genoeg is; ze moeten ook transparant zijn over de voortgang en afhankelijkheden. Ze zullen moeten aantonen dat ze de juiste tests hebben gedaan. Vertel het testverhaal, lever het bewijs.

Inbedden in de organisatie

De testuitdagingen worden technischer. Automatisering speelt een steeds grotere rol. Veel organisaties streven naar *continuous integration* en *delivery* en willen hun releasegang versnellen. Testen moet mee in deze versnelling en zal daarom geautomatiseerd moeten worden. Stubbing en servicevirtualisatie zijn manieren om de integratie van een component in samenhang met de andere systemen te testen zonder afhankelijk te zijn van de beschikbaarheid. Niet elk team heeft de kennis in huis om dit op eigen houtje te realiseren.

Tegelijkertijd zie ik het testvak verbreden. Het wordt steeds moeilijker om de



niet-functionele tests voor bijvoorbeeld beveiliging, performance en ux te negeren. Het testen van iot- en mobiele toepassingen vraagt om andere kennis dan nodig is voor het testen van de ons zo vertrouwde informatiesystemen. Dit maakt het erg lastig om alle tests te beleggen binnen het ontwikkelteam; dat heeft vaak niet alle kennis in huis.

De projectmanager die ik moest adviseren, wilde dat de value streams bleven werken. Integratie- of ketentests zullen nodig zijn, maar hoe om te gaan met deze teamoverstijgende verantwoordelijkheid? In de praktijk blijkt het te kort door de bocht om alles bij de teams te leggen zonder ze te ondersteunen en te faciliteren.

Sommige testuitdagingen overstijgen zelfs het it-domein. Organisaties beseffen meer en meer dat een minimaal levensvatbaar product alleen bedrijfsresultaat oplevert als ze de dienst ook daadwerkelijk kunnen leveren, het product kunnen verkopen of het proces kunnen ondersteunen. Het lijkt daarom logisch om ook businesstests uit te voeren. In plaats van geïntegreerde systemen te testen, moeten we de systemen inbedden in de organisatie. Ook *organizational readiness* wordt een testtopic, een businessperceel op de landkaart.

Waardevolle richtlijn

Aan de projectmanager legde ik dus uit dat als we agile werken en producten echt live willen brengen, we een aantal zaken moeten invullen. We zullen de percelen in kaart moeten brengen zodat we weten wat we getest moeten hebben om de commerciële

productlancering met vertrouwen tegemoet te kunnen zien. We zullen rekening moeten houden met verschillende kwaliteitsattributen en technieken. We zullen onze tests snel moeten doen en ervoor moeten zorgen dat we de resultaten kunnen verzamelen.

Daarnaast kunnen we de tests wel beleggen in de teams, maar moeten we hen misschien ook helpen als blijkt dat ze kennis of ervaring missen. Hiertoe kunnen we controlerende maatregelen nemen, bijvoorbeeld door audits uit te voeren om eventuele verbeterpunten te identificeren. We kunnen de teams helpen middels hands-on coaching op de werkvloer en we kunnen ze actief koppelen aan domeinexperts en specialisten voor inhoudelijke vragen. De keten- en businesstests dienen we onder te brengen ofwel bij de teams ofwel bij een

speciaal testteam. In het eerste geval zal het impact hebben op de voortgang binnen de teams en in het tweede geval ontstaan er afhankelijkheden die we moeten managen.

Ik keek de projectmanager aan en polste of hij de implicatie had begrepen. 'Er zijn zo veel testzaken te regelen op zo veel verschillende niveaus dat de kans klein is dat het vanzelf goed gaat', stelde ik. De teams zijn wel in de lead, maar er is ook een teststrategie nodig om de activiteiten op elkaar af te stemmen, de verantwoordelijkheden goed te kunnen invullen en de teamoverstijgende tests niet tussen wal en schip te laten vallen. Dit hoeft geen dik en lijvig document te zijn, als het maar de juiste dingen benoemt. Verder zullen we de governance moeten inrichten zodat we inzicht hebben in de voortgang van de tests en het project.

De meeting eindigde met het besluit om samen met de teams een agile teststrategie op te stellen en te implementeren. Tijdens het project is deze strategie een waardevolle richtlijn gebleken waarlangs activiteiten konden worden uitgezet, en daarmee een belangrijke succesfactor.



Derk-Jan de Groot (derkjandegroot@valori.nl) is testexpert en Agile-coach bij Valori.

Hij zet zich in om de toegevoegde waarde van softwareontwikkeling te vergroten in zowel Agile- als traditionele context. Hij is auteur van het boek 'Agile in de echte wereld - Starten met Scrum', dat eind oktober is verschenen bij uitgeverij Techwatch.

Redactie Nieke Roos