

Over het algemeen wordt weinig flexibel gedacht over het toepassen van agile technieken. Je gebruikt Scrum of XP of een andere methode en legt daar het handboek naast om het je eigen te maken. Je koopt een uitgebreide toolingset en dan kun je aan de slag. Helemaal fout, vindt Mike Jones van Outsystems. Zijn bedrijf heeft ruime ervaring met grote en kleinere agile projecten. En dat levert een steeds helderder benadering van agile op. Flexibiliteit is daarbij het sleutelwoord.

Agile: Klein starten, maar groot denken

Mike Jones: ‘Vertrouwen is basisprobleem’

Je moet agile terugbrengen tot een paar fundamentele zaken. Met kleine iteraties werken. Feedback verzamelen en bepalen wat de volgende stap wordt. Continue betrokkenheid van de business tot stand brengen. Je hoort wel eens zeggen dat de projecteigenaar niet nodig is. Dat is vanuit ons standpunt ondenkbaar. Als de business niet achter het project staat en niet voortdurend feedback geeft kun je niet agile werken. Je moet het project altijd kunnen bijstellen”, stelt Jones nadrukkelijk.

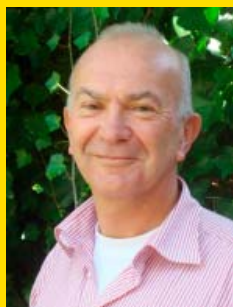
Jones probeert bepaalde aspecten van agile buiten de deur te houden, die het werken moeilijker maken. Maar de betrokkenheid van de klant, de continue feedback en na korte iteraties het project kunnen bijstellen zijn absolute voorwaarden om het te laten slagen. “We zijn voortdurend bezig om de klant bij het project te betrekken. Er zijn klanten, die het van het begin af aan begrijpen, maar de meesten zijn nog niet thuis in deze manier van werken en die proberen we er zoveel mogelijk bij te betrekken. Het vraagt nogal wat van het ontwikkelteam en de klant om iedere twee weken een deel van het project te kunnen opleveren. Dat geeft soms best veel stress voor de gemiddelde ontwikkelaar. We bestuderen hoe je het die ontwikkelaar zo werkbaar mogelijk

kan maken. En dat de business krijgt wat ze wil hebben. Daar gaat het uiteindelijk allemaal om”.

Agile is voor Jones een manier om het team te organiseren en een proces om effectief datgene te doen wat de business wil. Dat klinkt simpel. Je zou denken dat iedereen het kan. Maar er zijn wel een paar hindernissen te slechten. Die gaan over de skills, de tools en de ‘change’, de bereidheid om continu wijzigingen in de requirements te aanvaarden.

“Als je kijkt hoe organisaties hun agile inrichten, hebben ze in het begin vaak de neiging om een klein project op te pakken, ze gebruiken de bestaande tools en de beste mensen. Dat eerste project loopt meestal dus heel goed, omdat alle benodigde randvoorwaarden aanwezig zijn. En dan komt het volgende project. Vanwege het succes met het pilotproject durf je wel iets groters aan te pakken. Maar van de goede ontwikkelaars uit het pilotproject worden er een paar op een ander project gezet en er komen een paar andere zonder ervaring met agile voor in de plaats. De projectowner stuurt andere medewerkers, die wellicht ook geen ervaring met agile hebben”.

“Kortom: het proces is hetzelfde, daar heb je al enige ervaring mee. De tools blijven dezelfde, want die deden het in het pilotpro-



Robert de Ruiter
is hoofdredacteur
van Release Magazine.

ject ook goed, het skill level ligt vanwege het vernieuwde team lager, de complexiteit van het project wordt groter, er komen vanwege de omvang van het project meer wijzigingen in de requirements. Daar kun je flink mee in de problemen komen”.

Veel softwarehuizen kunnen de nieuwe omstandigheden wel aan. Tijdens het proces kunnen andere tools worden aangeschaft, de werkwijze kan worden aangepast. Er kunnen meer middelen worden ingezet. Je hebt als producteigenaar een budget. Maar je maakt je geen zorgen over het aantal releases dat er in een jaar worden vrijgegeven.

“Anders ligt dat in de corporate IT”, stelt Mike Jones. “Daar draait het vaker om het financiële verhaal. De vraag ‘wat gaat dat kosten’ staat centraal en als je teveel moet

beknibbelen gaat dat ten koste van het project. Dan is het zaak dat je met behulp van de juiste tools snel leert inschatten wat een project gaat kosten. Een andere vraag is: ‘hoe ga je met de veranderingen in het project om’. De requirements worden flexibeler, de verwachting is dat wanneer de business dat onderkent, ze ook met nieuwe ideeën komt. Maar als je in het team zit dat de code moet schrijven is het niet eenvoudig. Als je goed

bent kun je de code nog wel ombuigen om aan de nieuwe requirements te kunnen voldoen. Ben je als ontwikkelaar minder goed, dan druk je op delete en begint opnieuw. Dat kost natuurlijk heel veel tijd”.

Een goed framework voor de ontwikkeling is erg belangrijk om de wijzigingen te kunnen beheren. Als je bijvoorbeeld met Eclipse de veranderingen managet wordt de kans van slagen veel groter. En dan nog: de realiteit is dat geen enkel project loopt zoals je verwacht. Je kunt nog zo veel analyseren en documenteren – het datamodel, het proces, de procedures en noem maar op – maar uiteindelijk krijg je altijd met meer veranderingen te maken dan waar je op had gerekend. Het ontwikkelproces kost tijd. Ondertussen verandert de business en gaat lopende het proces andere eisen aan het project stellen. Het komt er op neer dat je eerst en vooral ingericht moet zijn op het doorvoeren van veranderingen. Flexibiliteit in het team en het proces zijn belangrijke items wanneer je agile wilt ontwikkelen. En dat geldt ook voor het hanteren van de gebruikte agile methode. Je kunt die zo naar je hand zetten dat je er optimaal mee kunt werken.

Het is in agile projecten vaak moeilijk om feedback te krijgen. Waar haal je die vandaan? Er kan een formeel team bij de klant zitten, maar soms ook ben je aangewezen op de helpdesk van de klant, waar de vragen over de software binnen komen. Bij een ander project sturen de gebruikers e-mails, waarin ze beschrijven welke problemen ze tegenkomen. Dit leidt vaak tot dubbelzinnigheid en het kost veel tijd om de problemen op te lossen. Door een duidelijk proces op te zetten om feedback te krijgen kun je hier veel winst boeken.

In de corporate IT is vaak niet duidelijk waar de richtlijnen voor de applicatie vandaan komen. Softwarehuizen hebben productmanagers, die zich een duidelijk idee hebben gevormd over wat de markt wil. In de business is het opstellen van de richtlijnen vaak een parttimebezigheid van een IT-medewerker, die opdracht heeft gekregen ‘iets nieuws’ te ontwikkelen. Hij geeft de opdracht door – vaak met onduidelijke



Mike Jones: “Niet te hard van stapel lopen”.

requirements – de code wordt geschreven, het resultaat getoond en dat is dan teleurstellend.

Vertrouwen

“Vertrouwen is een basisprobleem bij agile. Om een project succesvol te kunnen afronden moet je soms rigoureus te werk gaan. We brengen alle partijen bij elkaar om aan het begin van het project tekst en uitleg te geven over de werkwijze. We stellen ook precies de datum en de tijd vast waarop we iteraties opleveren. We houden ons streng aan dat schema. Je ziet dan dat tegen de tweede of derde oplevering het vertrouwen begint te groeien. De klant had meestal namelijk niet verwacht dat we inderdaad op tijd zouden opleveren. Het heeft nog een ander positief

effect. De business raakt op deze manier meer betrokken bij het project, gaat uitzien naar de volgende oplevering. Je krijgt ook meer feedback, meer begrip voor het feit dat er wel eens iets mis gaat, waarvoor je later meer tijd moet inruimen”, aldus Jones.

“Van alle bekende methodes komt Scrum nog het dichtst bij de ware bedoeling van agile werken. Maar wij gebruiken het niet volgens de letter. We hebben bijvoorbeeld

‘Aan het einde ontstaat wellicht de wens om het proces bij te stellen. Maar loop in het begin niet te hard van stapel’


de rol van de scrummaster in twee verschillende rollen gesplitst. Bij alle projecten, die wij voor klanten doen, werken we op afstand. We hanteren daarom een soort nearshore/offshore benadering. Dit is in onze methode vastgelegd. Ook houden we aan het eind van het proces een tuning-sprint. Wij timeboxen niet alleen de sprints, maar het hele project. Aan het eind, wanneer de applicatie in gebruik wordt genomen, plannen we deze tuning-sprint. Tot dat moment hebben alleen de IT-ers van de klant het project gevolgd. Maar bij de tuning-sprint betrekken we een veel groter aantal gebruikers. Dat zijn er soms twintig, soms ook wel een paar honderd”.

“Het developmentteam blijft aanwezig bij de ingebruikname van de software. Het bepaalt de prioriteit van de wijzigingen, zorgt dat deze op tijd worden afgerond, enzovoort. Soms zijn het maar kleine dingen. Een knop die op de verkeerde plaats zit. Soms ook moet meer code worden geschreven. Het psychologische voordeel: aan het eind van deze tuning-sprint zijn de gebruikers ook een beetje ‘eigenaar’ van het project geworden. Ze hebben van nabij meegemaakt dat aan hun wensen tegemoet wordt gekomen, dat er daadwerkelijk wijzigingen worden doorgevoerd om aan hun wensen tegemoet te komen. De adoptie van de applicatie wordt daardoor veel eenvoudiger. Dit wordt niet omschreven in het scrumproces, maar wij merken dat het heel goed werkt”.

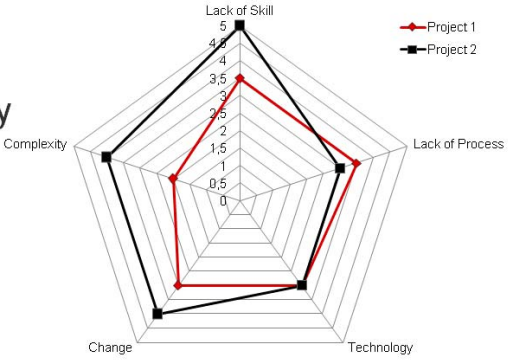
Klein beginnen

Mike Jones sluit af met een paar aanbevelingen. “Heel belangrijk als je met agile van start gaat is ‘start small, think big. Met start small bedoel ik dat je gaat werken met de tools die in huis zijn. Koop geen nieuwe. Maak je niet druk over de methode die je gaat gebruiken. Of dat XP of Scrum is maakt niets uit. Kijk vooral wat je kunt gebruiken. Er zijn immers geen twee bedrijven gelijk, dus wat in de ene situatie handig is hoeft dat in een andere niet te zijn”. “Uiteindelijk krijg je steeds meer inzicht in- en ervaring met een bepaalde methode en met de tools. Dan ontstaat wellicht de wens om het proces bij te stellen of te automatiseren. Maar loop in het begin niet te hard van stapel”. «

Scaling Agile
The Challenge




- People and skills
- Agile process
- Delivery technology
- Amount of change during the project
- Overall project complexity

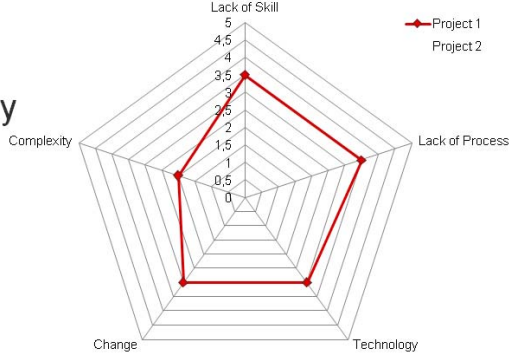


Why Second Projects Fail

Scaling Agile
The Challenge



- People and skills
- Agile process
- Delivery technology
- Amount of change during the project
- Overall project complexity



In het tweede project wordt de complexiteit groter, de ervaring van het team minder en de veranderingsbereidheid kleiner. Daarom gaan vaak tweede projecten de mist in.