

**Scott W. Ambler is net als Jeff Sutherland een van de kopstukken uit de Agile wereld. Door zijn columns in Dr. Dobbs, zijn boeken en zijn vele lezingen heeft hij grote bekendheid verworven. Hij heeft het Agile Unified Process ontworpen, een vereenvoudigde versie van RUP en het Enterprise Unified Process, een uitbreiding op het Rational Unified Process. Ook is hij Practice Leader Agile Development bij IBM.**

Scott Ambler:

## Niet iedereen is geschikt voor Agile

*Toen Scott Ambler eindelijk verscheen voor het interview was het al happy hour. Toepasselijk, aangezien het in een café plaatsvond. Ambler is één van die mensen die je nauwelijks hoeft te interviewen, één vraag levert meteen een pagina op en met een paar vragen heb je al een publiceerbaar verhaal. Maar Ambler – misschien ook een beetje geholpen door ons eigen Heineken – bleef doorpraten, ook na mijn laatste vraag, dus uiteindelijk zijn het twee interviews geworden. Een deel van het verhaal dat meer op RUP, Jazz en grote projecten ingaat is te lezen in Software Release Magazine 5/2008, en is voor de lezers van Java Magazine spoedig te lezen via het online archief. Het gedeelte van het gesprek dat plaatsvond het uur voordat Ambler in de taxi stapte, ging helemaal niet meer over IT en hoewel het zeker amusant was hebben we het maar niet gepubliceerd. Het lange gesprek voor Java Magazine begon echter met de vraag wanneer je wél, en wanneer je niet Agile doet.*

Ambler: “Er zijn veel goede redenen om het wel te doen. Anders dan veel mensen denken brengt Agile werken minder risico’s met zich mee dan traditioneel ontwikkelen. Het levert ook betere kwaliteit op en er is meer feedback met de business dan traditionele software. Als de business-gemeenschap echt de implicaties van Agile zou begrijpen, hoe het meer controle

aan hen geeft, meer transparantie, dan zouden ze het gewoon eisen. Traditionele projecten zijn zo weinig transparant, het is zo gemakkelijk om dat wat er gebeurt in een statusrapport anders voor te stellen. Je kunt documenten schrijven en ervoor zorgen dat ze geaccepteerd en ondertekend worden, maar volgen mensen die ook? Je weet het nooit. Inherent daaraan weet de business-gemeenschap dat het soms heel moeilijk voor hen is om uit te vinden wat er echt gebeurt in de IT-afdeling en Agile opent het allemaal. Dat is natuurlijk ook een probleem als je niet aan de wensen van de business kunt voldoen. Agile schaaft zeker, portfoliomanagement is ook veel gemakkelijker te beheren dan in de traditionele wereld. Het is aantrekkelijk voor veel ondernemingen. “

*Maar wanneer doe het je het niet?*

Ambler: “Agile is niet direct gerelateerd aan de technologie, het gaat er om of je mensen de vaardigheden hebben om het te doen. Als je een organisatie hebt waar je mensen flexibel zijn en samenwerken dan zijn er weinig dingen waarbij ik het niet zou toepassen. Maar er zijn ook projecten waarbij men zich minder gemakkelijk voelt bij Agile. Ik weet dat Agile betere kwaliteit oplevert en voor *life critical* systemen zou het mijn keus zijn, maar realistisch gezien zullen de meeste organisaties waterval eisen, nog steeds

aandringen op het weggooien van veel geld omdat het wordt gezien als minder risicovol dan *Agile*. Misschien is het dat voor hen ook wel, afhankelijk van de aard en de vaardigheden van hun personeel. Bij *Agile* test je veel meer en veel eerder, je richt je ook meer op kwaliteit en dus krijg je ook een hogere kwaliteit dan in de traditionele wereld. Ik heb nooit de Space Shuttle gebouwd of een vliegtuig. Er zullen een paar klassieke problemen zijn die je niet met *Agile* kunt oplossen. Een issue waar ik wel naar kijk is het deployment. Als ik honderdduizend cd's verstuur en ik ontdek een fout, dan is dat een serieus probleem. Of als ik een satelliet de ruimte in stuur en ik kan de software niet updaten, of in het algemene dingen die heel moeilijk veranderd kunnen worden. Ik zou ertoe geneigd zijn wat meer serieel te gaan werken bij dat soort opgaven. Maar dan zou ik nog steeds zo goed mogelijk

iteraties doen, veel samenwerken en testen. Je kunt wel serieel werken maar dat betekent niet dat je er stom mee om hoeft te gaan. Je kunt veel van de ideeën uit de *Agile* gemeenschap overnemen en toepassen, maar dan toch volgens een seriële, waterval lifecycle werken om welke reden dan ook."

*Veel mensen zijn er wel van overtuigd dat Agile de juiste weg is, maar in de praktijk blijkt dat ze project na project op de klassieke manier doen. Hoe is zoiets mogelijk?*

Ambler: "Ja, dat hoor je heel veel. Wat er vaak gebeurt is dat de IT-mensen wel *Agile* willen werken. Maar wanneer de business nog steeds alle requirements vooraf vast wil timmeren, met de zogenaamde accurate estimates van te voren, dan wordt het min of meer onmogelijk voor de IT om effectief te zijn. IT en business moeten dezelfde kant op gaan.

Anders kan de business beslissingen nemen waarvan ze denken dat ze helpen om financiële risico's in te perken, maar die - als ze zich de IT-implicaties ervan niet kunnen voorstellen - heel anders uitpakken. Een heel goed voorbeeld is een gedetailleerde begroting aan het begin van een project. De business vraagt daarom, omdat ze hun financiële risico willen inperken. Ze realiseren zich echter niet dat dit het financiële risico niet verkleint, maar juist vergroot; ze begrijpen immers niet de implicaties van wat ze doen op een hoger niveau. Om eerlijk te zijn is dat ook wat hen sinds jaar en dag gevraagd is. Om welke reden dan ook zijn ze niet in staat te zien dat dingen anders lopen dan ze verteld is. Ze vragen om een exacte begroting, krijgen er een van één miljoen en dan bedragen de kosten anderhalf of juist maar driekwart miljoen. Om de een of ander reden zijn ze niet in staat te zien dat dit niet goed werkt. Bij het volgende project vragen ze weer om een exacte begroting. Je kunt een beetje rommelen en net doen alsof maar je echte goede begroting wil maken, maar daarvoor heb je heel veel informatie nodig. Maar als je die verzamelt schrijf je uiteindelijk software naar voorgegeven specificaties. Omdat we echter niet goed zijn in het voorspellen van wat we eigenlijk willen, bouwen we uiteindelijk iets wat we niet willen."

### **On time, on budget, on spec**

Onderzoek van de Standish Group, Gartner en anderen laat zien dat de overgrote meerderheid van de projecten niet op tijd, op budget en volgens de specificaties geleverd wordt. Dr. Dobb's heeft bijna een jaar geleden een onderzoekje gedaan onder beslissers naar de vraag hoe mensen project-succes definiëren.

Ambler: "Meestal wordt aangenomen dat het antwoord op die vraag luidt: een systeem dat op tijd, op budget en volgens de specificaties geleverd wordt, maar dat is niet waarin ze geïnteresseerd waren. Ze wilden waar voor hun geld, ze wilden software die hun behoeften vervult, software van goede kwaliteit die afgeleverd wordt wanneer hij echt klaar is in plaats van volgens het schema. Hoewel ze dus iets heel anders willen, zullen bij het volgende project wéér eisen dat het *on time*, *on budget* en *on spec* is. Dat gebeurt om een aantal redenen. In de eerste plaats is hen verteld dat te doen. In de tweede plaats zijn ze zich er niet van bewust dat



ze betere keuzes hebben. De *Agile* mensen werken eraan om de juiste dingen te bouwen, van hoge kwaliteit en op een efficiënte manier, maar de business gelooft niet dat we daartoe in staat zijn. Dat is echt een probleem. Een ander probleem is dat system integrators vaak gedwongen worden om in te schrijven op basis van een exacte begroting en daarmee doen ze zichzelf tekort. Daar zit in de eerste plaats de klant achter, omdat die – ten onrechte – denkt op die manier de meeste waarde voor zijn geld te krijgen. Alle SI's zijn daarover gefrustreerd. Ik dring er nu op aan dat ze tegen de wensen van de klanten in gaan en zeggen: 'Ja, we kunnen volgens dit proces te werk gaan, die opdracht zo vervullen als jullie dat vragen, maar het is geen goed idee. Het is niet goed voor jullie, niet voor ons, en er is een betere manier van werken: de *Agile* manier.' Je kunt dat doen als je een vertrouwensrelatie hebt. Maar als ze je niet goed kennen of ze het per se op de bekende manier willen doen, dan helpt het niet."

### IT-industrie veranderen

Ambler hoopt dat wanneer de IT-gemeenschap in zijn geheel hier tegenin gaat, de business uiteindelijk zal begrijpen dat de traditionele manier van werken tegen hun eigen belangen in gaat.

Ambler: "Nu gaan de IT-ers er gewoon op de eisen van de business in en zeggen: 'dat zijn de mensen met het geld en die hebben het voor het zeggen'. We zullen dit probleem niet morgen kunnen oplossen, maar over een periode van tien jaar kunnen we gehele IT-industrie veranderen, zonder enige twijfel. We moeten dan wel besluiten dat we het willen veranderen. We moeten ook veel beter worden in het begrijpen van de business-kant van de zaken. IT-ers zijn daar niet goed in, ook al doen we alsof."

*Dus er is heel wat werk aan de winkel.*

Ambler: "Ja, en om eerlijk te zijn: de business is tamelijk boos op ons, want we zijn veel beloften niet nagekomen. We brengen al die nieuwe technologieën en zijn voortdurend dingen aan het veranderen. Zij willen gewoon dat het werk gedaan wordt. Er zijn veel uitdagingen daar en ze hebben eigenlijk gelijk wanneer ze boos zijn."

*Hoe schaalbaar is Agile, er wordt daar de laatste tijd veel over gediscussieerd.*

Ambler: "De belangrijkste boodschap is dat *Agile* schaalbaar is. Je moet wel kijken naar het grote plaatje. Het zijn niet alleen de grote teams waar je je zorgen over moet maken, ook of het gedistribueerd over meerdere locaties, regels met betrekking tot veiligheid of andere zaken, andere partijen die erbij betrokken zijn, je legacy-systemen. Soms zijn er echt vervelende beperkingen, heb je een governance-programma, er zijn veel issues die je niet weet."

*Wat vindt u van UML?*

Ambler: "Ik ben er helemaal niet tegen, maar je moet je realiseren dat een heel groot deel gewoon op een bord getekend wordt, en of dat nu UML is of niet, dat kan me eigenlijk niet schelen. Je communiceert, je onderzoekt ideeën, daar gaat het om. Maar nu heb ik veel ervaring met modelleren en met de tools. Veel ontwikkelaars hebben die ervaring niet. Ik zou ze willen aanraden ervoor te zorgen dat ze die wel krijgen. Natuurlijk staat het goed in je CV, maar afgezien daarvan kun je ook je productiviteit enorm verhogen. Ik doe bijvoorbeeld veel database-werk. Ik zou van zijn leven geen DBL schrijven, ik modelleer het liever en druk op een knop. Waarom zou ik Java scaffolding code schrijven? Tools genereren een hoop voor je. Waarom zou ik het schrijven wanneer ik het met het trekken van een paar lijntjes kan genereren? Als ik programmeer dan gebruik ik de juiste tool voor het werk, maar daarvoor heb ik wel modelleervaardigheden nodig. Je hebt tijd nodig om te leren modelleren, en om met de tools om te leren gaan. Je moet visueel kunnen denken, niet iedereen kan dat."

### Modelleervaardigheden

Jammer genoeg zijn er niet zoveel mensen met modelleervaardigheden dan er zouden kunnen zijn. Maar een van de redenen dat mensen zo teleurgesteld zijn in modellen, is dat in de traditionele wereld er vaak een grote breuk is tussen de modelleerinspanningen en de codeerinspanningen. Er wordt veel te veel tijd gestopt in het seriële modelleren. Wanneer je iteratief modelleert dan zie je dat modelleren een hoop oplevert, maar in serieel werk veel minder. Ik zie steeds voor me hoe heel slimme designers en architecten heel gedetailleerd modelleerwerk doen en het aan de gewone ontwikkelaars geven..."

*Die er dan vervolgens niet meer naar kijken!*

Ambler: "Ja, want óf de ontwikkelaar heeft niet zo'n hoog niveau en begrijpt er niets van, óf hij begrijpt het wel maar denkt dat hij het beter weet, óf er is eenvoudigweg geen tijd om het nog te implementeren. Wanneer de ontwikkelaars al zijn begonnen met het schrijven van code voordat ze die modellen zien, dan weet je al dat ze niet meer gebruikt worden. Dit gebeurt veel te vaak. Uiteindelijk wordt dan vaak de oplossing gezocht in het opleggen van modellen en dergelijke met een hoop bureaucratie. Je maakt dan een proces dat toch al niet goed verliep nog langzamer."

*Welke mensen zijn geschikt voor Agile, of beter gezegd, zijn sommige mensen er niet voor geschikt?*

Ambler: "De mensen die van *Agile* houden zijn flexibel, ze willen graag samenwerken, kunnen goed met mensen omgaan of hebben in ieder geval de potentie om dat te leren, en ze willen graag concrete feedback hebben wanneer ze software afleveren. Ze willen vooral leren, dat is misschien wel de belangrijkste voorwaarde. Degenen die er moeite mee hebben zijn vaak te zeer gespecialiseerd, willen in feite alleen maar één ding doen, alleen maar businessanalyse, programmeren of dba-werk en hebben nauwelijks of geen interesse om er andere dingen bij te leren. Ze willen hun werk doen en ze willen dat goed doen, maar dan willen ze het aan de volgende afgeven en dat is het dan. Dat lijkt effectief wanneer je naar hun kleine specialisme kijkt, maar wanneer je naar het grote geheel kijkt is het ongelofelijk ineffectief. Dat is waar al die bureaucratie om de hoek komt kijken, al dat extra papierwerk enzovoorts. Natuurlijk, we hebben dertig tot veertig jaar lang mensen gevraagd zo te werken en we hebben ze daarvoor beloond. Dat heeft ook een selectie met zich meegebracht. Nu komen we ineens aanzetten en zeggen: 'je moet je vaardigheden uitbreiden, je moet flexibel zijn en samenwerken'. Dat is een grote verandering voor veel mensen. Sommigen zijn daartoe in staat, anderen niet. Maar dat is niet erg, want er zullen altijd niet-*Agile* taken overblijven en daarnaast zal het voor veel organisaties misschien wel tien jaar duren voordat ze volledig *Agile* zijn. Bij sommige bedrijven zal het misschien nooit gebeuren. De traditionele mensen, die niet zo flexibel zijn, zullen het echter steeds moeilijker gaan krijgen." «