

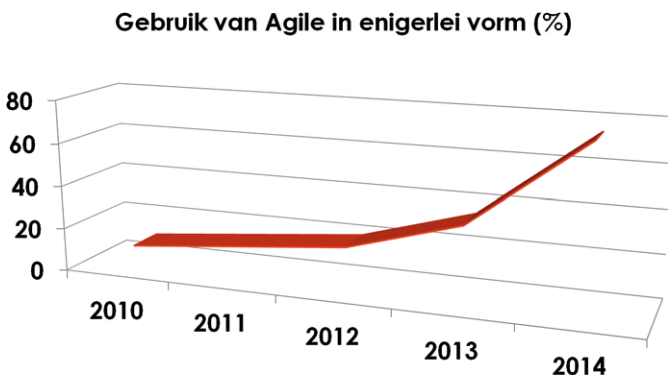
# Vijf jaar agile. Hosanna of Drama?

Leo van der Aalst – Fontys Hogeschool ICT, Sogeti

In dit artikel wordt een top vijf van vier onderwerpen op het terrein van agile werken geschetst: vijf agile misvattingen, vijf agile valkuilen, vijf agile voordelen en vijf agile succesfactoren. Deze zijn gebaseerd op vijf jaar ervaring als agile consultant en agile trainer in meer dan 300 agile projecten. Dit waren niet alleen projecten in Nederland maar ook in vele andere landen. Hoewel het ruim 300 subjectieve meningen zijn, geeft dit uiteindelijk een objectief beeld van de actuele situatie met betrekking tot agile projecten. Daar waren zowel Hosanna als Drama projecten bij.

## Algemene trends

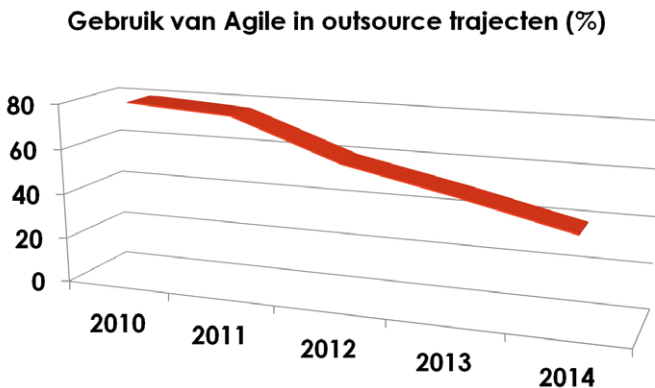
Agile aanpakken zijn al langer bekend, in ieder geval al vanaf 2001. Dat was het jaar waarin het agile manifeste met haar vier waarden en twaalf principes werd opgesteld door zeventien prominente softwareontwikkelaars. Toch heeft het daarna nog een tijd geduurd voordat agile aanpakken door IT-organisaties werden omarmd. Pas in de laatste vijf jaar is dit substantieel toegenomen. Het gebruik van een agile aanpak is van minder dan 20% in 2010 gestegen naar bijna 80% in 2014 (zie figuur 1).



Figuur 1: Gebruik van een agile aanpak in enigerlei vorm (%)

In 70% van de organisaties die een agile aanpak gebruiken, is dat scrum. Minder dan 15% van de organisaties gebruiken een andere aanpak, zoals Kanban, Feature Driven Development, Agile Unified Process, DSDM/Atern, Lean of Extreme Programming.

Een andere interessante trend is de afname van een agile aanpak in outsource trajecten. Van bijna 80% vijf jaar geleden, naar minder dan 40% nu (zie figuur 2).



Figuur 2: Gebruik van een agile aanpak in outsource trajecten (%)

Als gevraagd wordt naar de tevredenheid over het resultaat van een agile project, dan zegt meer dan 50% tevreden te zijn. Dit wordt enigszins gekwantificeerd door een veronderstelde productiviteitstoename van 0 tot 50% met een gemiddelde van 20%, een kostenreductie van 0 tot 50% met een gemiddelde van 30% en een snellere time-to-market van 10 tot 60% met een gemiddelde van 30%.

## Agile Misvattingen

- Agile is het antwoord op alles
- In Agile ontbreken processen
- In Agile wordt niet gepland
- In Agile ontbreekt discipline
- In Agile wordt niet gedocumenteerd

*Agile is het antwoord op alles!* In ieder geval beter dan een traditionele softwareontwikkelaanpak, zo wordt gezegd. Helaas, in de praktijk blijkt dat niet zo te zijn. Iedere organisatie moet, misschien zelfs op projectniveau, een afweging maken welke aanpak te kiezen. Dat kan een agile aanpak zijn, maar in omgevingen met veel hiërarchie, vast omschreven procedures en processen, of enige vorm van certificering (o.a. luchtvaart, security) lijkt agile minder geschikt te zijn dan een traditionele aanpak.

*In agile ontbreken processen, in agile wordt niet gepland en in agile ontbreekt discipline!* Dit zijn wel heel vaak gehoorde misvattingen. In agile aanpakken bestaan, net als in traditionele aanpakken, wel degelijk processen. Dit zijn weliswaar lichtgewicht processen, die door het agile team worden aangepast aan hun eigen behoeften. En dan wordt er gezegd, iedereen doet maar wat, zonder planning en discipline. Het tegendeel is eerder waar! Ga maar na, hoe kun je in een iteratie van bijvoorbeeld twee weken werkende software opleveren zonder planning en discipline? Inderdaad. Niet dus! Release en iteratieplanningen maken vast onderdeel uit van agile aanpakken, waarbij scrum- of kanban borden, burndown charts en de daily scrum worden gebruikt om werkzaamheden transparant te maken en af te stemmen. Dit alles ondersteunt de planning om te komen tot het gewenste resultaat. En in plaats van het ontbreken van discipline is eerder een ijzeren discipline van de agile teamleden gewenst om dit resultaat als team binnen de iteratie te kunnen bereiken.

Een hardnekkige vijfde misverstand is dat er *in agile niet wordt gedocumenteerd*. In het algemeen wordt door het agile team alleen gedocumenteerd wat nodig is om als agile team het werk te kunnen doen of waar stakeholders een belang bij hebben. In het ene geval is dat minimaal, omdat de teamleden precies weten wat ze moeten doen en wat ze van elkaar kunnen verwachten. In het andere geval kan dat meer zijn, omdat er wellicht eisen zijn gesteld aan bijvoorbeeld onderhoudbaarheid of overdraagbaarheid van producten. Of omdat er wellicht een wet- en regelgeving-, of certificeringseis is, die meer documentatie vraagt dan voor het team nodig was om het product te maken.

## Agile Valkuilen

- Overboord gooien van alle bestaande software ontwikkelprocessen
- Gedeeltelijke adoptie
- Programmeren tot de laatste minuut
- Korte termijn denken (→ technical- / test debt)
- Alles verandert, behalve .....ik

Je zult maar als IT-er al jaren met een bewezen goedwerkende softwareontwikkelaanpak werken en dan beslist het management op een goede - of kwade? - dag dat er nu agile gewerkt moet gaan worden en dat, 'by the way', die goedwerkende aanpak niet meer kan worden gebruikt. Het *overboord gooien van bestaande softwareontwikkelprocessen* is een vaak voorkomende valkuil. Want wat staat er nu helemaal in het agile manifeste of in de scrumaanpak over hoe je software moet ontwikkelen? Over hoe je requirements, functionele specificaties

als user stories of use cases, technische specificaties, datamodellering, systeem- en software architectuur, software, tests, moet opstellen en maken? Niets toch? Dus dan lijkt het verstandig niet het kind met het badwater weg te gooien, maar goedwerkende onderdelen van de softwareontwikkelaanpak – en testaanpak - te blijven gebruiken of aan c.q. in te passen aan de agile situatie.

Vrijwel alle methoden zijn goed doordacht en werken prima als ze worden gebruikt zoals ze zijn bedoeld. Denk aan SDM, JSD, RUP of PRINCE2. Bij de implementatie van een agile aanpak als scrum is vaak een *gedeeltelijke adoptie* te zien. Even zo vaak blijkt dat zowel een bewuste als een onbewuste keuze te zijn. Zelfs een bewuste keuze door middel van 'cherry picking' leidt in de meeste gevallen tot – veel – minder succes dan een volledige adoptie. Het bekende gezegde 'hinken op twee gedachten' doet hier opgang en is een notoire valkuil.

*Programmeren tot de laatste minuut.* Wie komt dat niet bekend voor? Dit is een diepe valkuil, want dit betekent per definitie dat er geen werkende software aan het eind van een iteratie wordt opgeleverd. Alle agile aanpakken zeggen, in iets van elkaar afwijkende bewoordingen, dat een product pas gereed ('done') is, als het is getest en is geaccepteerd ('ge-demo't') door de stakeholders. Daarbij brengt het *korte termijn denken* het risico op test- en technical debt met zich mee. Door wel snel 'iets' - omdat bijvoorbeeld de 'product owner' dat wil - op een 'quick and dirty' wijze aan het eind van de iteratie op te leveren, kunnen code-, architectuur- en testgerelateerde onvolkomenheden in volgende iteraties aan het licht komen, die tot grote, kostbare of tijdrovende herstelactiviteiten kunnen leiden.

En ineens zit je niet meer in een traditioneel ontwikkelproject, maar in een scrumteam. Nu maar hopen dat de andere teamleden hiermee kunnen omgaan, denk je. Dit is een heel typische valkuil, *alles verandert, behalve ..... ik!* Een menselijke eigenschap is dat mensen al gauw van zichzelf vinden dat ze iets wel kunnen, terwijl ze denken dat andere mensen daar meer moeite mee hebben. Maar in scrum is het van belang dat je zelf ook mee verandert. Dat gaat niet vanzelf, daar moet je echt iets aan doen.

## Agile Voordelen

- Projecten zijn transparanter
- Prioriteiten kunnen makkelijker worden bijgestuurd
- Kwaliteit producten is beter
- Kortere time to market
- Medewerkers zijn gemotiveerder

'Burndown charts', scrumborden met user stories, taken en kolommen, die laten zien of een user story bijvoorbeeld 'in progress' of al 'done' is. De actuele status van een scrumproject is vaak in één oogopslag te zien. In het algemeen kun je zeggen dat *agile projecten transparanter zijn*. En door de korte iteraties, de nabijheid van de stakeholders - via de 'product owner' - en de nauwe samenwerking in het scrumteam kunnen *prioriteiten makkelijker worden bijgestuurd*, waardoor sneller - op wat voor verandering dan ook -, kan worden gereageerd.

De *kwaliteit van producten is beter*. Iedere IT-er kent wel de figuur met de schommel, die laat zien hoe de wens van de klant uiteindelijk – afwijkend – wordt gerealiseerd. Onderzoeken laten zien dat in traditionele omgevingen van de oorspronkelijk honderd gedefinieerde requirements er vijftig worden gerealiseerd, waarvan er uiteindelijk maar vijftig door de klant worden gebruikt. Dat is nogal wat 'waste', volgens Lean begrippen. Maar in een scrumproject zitten de stakeholders, zoals in vorige alinea beschreven, dicht op het ontwikkelteam. Een afwijking op het gewenste product wordt snel gezien en kan net zo snel worden hersteld. Ook als er nieuwe wensen zijn, of juist wensen die zijn vervallen, kan er snel worden ingegrepen, waardoor het gerealiseerde product (heel) dicht bij het gewenste product komt. Een ander voordeel is dat de '*time-to-market*' in een agile project vaak met 10 tot 60% wordt verkort. Met andere woorden, de juiste producten worden ook nog eens sneller productieklaar opgeleverd!

Vaak leven uitgebluste IT-ers helemaal op als ze in een agile project gaan werken. Of dat nu komt door het verlaten van een jarenlange sleur, of door het werken in een team aan een gezamenlijk doelstelling, of doordat het team meer verantwoordelijkheid heeft? Feit is, dat in agile projecten *medewerkers vaak gemotiveerder* zijn. Ze zijn gewoon ook blijer. Een niet te onderschatten emotie voor het behalen van goede resultaten.

## Agile Succesfactoren

- Bedrijfsbrede commitment is vereist
- Invoering heeft begeleiding nodig
- Mensen moeten worden vertrouwd
- Team moet acteren als team
- 'Fail to succeed'

Ja managers, nu komt het er op aan. In het verleden was het zo, dat een team of de medewerkers de beslissingen van de manager vertrouwde en die opvolgden. Nu, in de agile aanpakken, is het de manager die de beslissingen van het agile team moet accepteren. Het is de manager die *de mensen in het team moet vertrouwen* en aan hen zoveel mogelijk vrijheden en (beslis)bevoegdheden moet geven. Om

écht succesvol te kunnen zijn met de invoering van een agile aanpak is *bedrijfsbreed commitment* nodig. De invoering vereist meestal een cultuuromslag of minimaal een culturaanpassing, waardoor het bijvoorbeeld heel gebruikelijk is dat mensen die niet in een agile omgeving kunnen werken het bedrijf verlaten en andere mensen daarvoor in de plaats komen.

Ook voor de teamleden komt het er nu op aan. Succes wordt alleen bereikt als het *team daadwerkelijk als team acteert*. Dus niet werken in silo's van ontwerpers, programmeurs, testers en gebruikers. Maar allemaal samen. Verder moeten alle ontwikkelprocessen worden geïntegreerd. Dus bijvoorbeeld geen losstaand ontwikkel- en testproces, maar volledig geïntegreerd gelijktijdig ontwerpen, programmeren en testen. Deze integratie geldt ook voor alle teamleden en activiteiten. Iemand in een programmeursrol moet bereid zijn andere activiteiten op te pakken, bijvoorbeeld testactiviteiten. Iemand in de testrol moet ook bereid zijn bijvoorbeeld testautomatiseringsactiviteiten op te pakken of de product owner te helpen bij het opstellen van de user stories. Dit alles is uiteraard afhankelijk van de kennis en kunde van de desbetreffende persoon. In het algemeen kun je zeggen, ieder teamlid beheerst één discipline uitstekend en is bereid een tweede of een derde bij te leren. Teamleden in een agile team zijn multidisciplinair.

Zoals uit bovenstaande alinea's blijkt, heeft de invoering van een agile aanpak nogal wat voeten in de aarde. Er zijn veel organisaties die daar mee worstelen, om niet te zeggen, maar wat aan modderen. Zeker ook omdat er, vooral bij bepaalde managers, weerstand tegen de verandering bestaat. De echte 'key' succesfactor is dan ook voor het zorgen van *begeleiding bij de invoering* van een agile aanpak. Bijvoorbeeld in de persoon van een veranderingsmanager of scrum coach.

De invoering van een agile aanpak is een leertraject en gaat gepaard met vallen en opstaan. De ene keer is de invoering sneller succesvol dan de andere keer. Het is belangrijk om telkens te leren van de gemaakte fouten en die in bijvoorbeeld een volgende iteratie weer te verbeteren. Dus niet te snel opgeven. Wees ook niet bang fouten te maken. Want hier geldt, *'fail to succeed'*!

## Conclusie

Thomas Edison heeft ooit gezegd, "I haven't failed. I've just found 10.000 ways that won't work."

In dit artikel vindt u hopelijk voldoende inspiratie om een eventueel Drama project in een Hosanna project te veranderen. Of nog beter, direct een Hosanna agile project te starten! Waardoor u niet op poging 10.001 hoeft te wachten!