



Agile en IT-contracten – een paradox?

Introductie

Voor veel IT-organisaties en leverancier is agile ontwikkelen de standaardaanpak: voor hen is het de meest effectieve manier om een project te realiseren. Voor IT-juristen geldt dit verbazingwekkend genoeg niet. Zij schrijven contracten en voorwaarden waar het begrip agile geheel niet in voorkomt en waarin geen rekening gehouden wordt met de verschillen tussen agile en een klassieke manier van werken. In een kort onderzoek is onderzocht waar deze paradox tussen de juridische en de technische IT-wereld op is gebaseerd, en hoe men tot een goed contract voor een agile project kan komen.

Achtergrond en opzet van het onderzoek

Voor dit onderzoek zijn 15 interviews uitgevoerd met in IT gespecialiseerde advocaten (zie de appendix voor een overzicht van betrokken kantoren). Er is bewust gesproken met in IT gespecialiseerde advocaten, van zowel grote als kleinere kantoren om een compleet beeld te krijgen. In de interviews is gevraagd naar de praktijkervaring met agile, de risico's die zich voordoen in IT-projecten met en zonder agile aanpak, en welke aandachtspunten men belangrijk acht in een IT-ontwikkelcontract. De resultaten uit deze interviews zijn aangevuld met de ervaringen van de auteur uit de agile IT-praktijk, om zo een goed beeld te krijgen van de tegenstellingen tussen de werelden van IT-uitvoering en IT-contracten.

De uitkomsten van het onderzoek zijn verwoord in de vorm van antwoorden op een aantal specifieke vragen, die als relevant naar voren kwamen uit het interview:

- Wat betekent agile precies?
- Wordt agile in de praktijk gebruikt (oftewel: hoeveel van de projecten wordt volgens agile gebruikt?)
- Wat agile aangeraden vanuit juridisch perspectief?
- Hoe maak je een goed IT-ontwikkelcontract? (aanbevelingen voor een goed agile-contract die ook gelden voor een niet-agile ontwikkelcontract)
- Wat zijn specifieke agile contractafspraken?

Vraag 1: Wat betekent agile precies?

De term agile (letterlijk: wendbaar) wordt door veel advocaten, samen met de term cloud, gezien als een mode-kreet. Het is een breed begrip dat vaak gebruikt wordt zonder dat het een concrete invulling

krijgt. Dit klopt eigenlijk wel: de term agile is een verzamelterm. De term is gedefinieerd in het agile manifest, dat in 2001 is opgesteld. Het Manifesto bestaat uit vier hoofdregels en 12 principes. De hoofdregels zijn eenvoudig:

1. **Mensen en hun onderlinge interactie** boven processen en tools
2. **Werkende software** boven allesomvattende documentatie
3. **Samenwerking met de klant** boven contractonderhandelingen
4. **Inspelen op verandering** boven het volgen van een plan

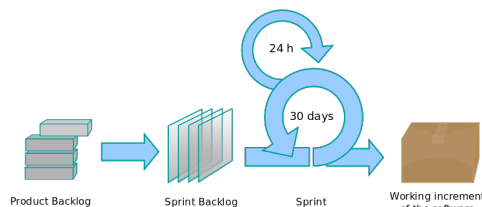
Met name de derde en de vierde regel zijn juridisch spannend: doordat klant en leverancier samenwerken, is de klant medeverantwoordelijk voor een goed of slecht eindresultaat. En inspelen op verandering betekent dat het oorspronkelijke plan niet heilig is, maar tijdens het project kan worden aangepast.

De hoofdregels en bijbehorende principes leggen vooral bedoelingen vast, maar worden niet concreet uitgewerkt. Er zijn verder geen dwingende regels. Agile is dus geen concrete methode en er zijn geen voorgeschreven elementen. ‘Agile’ is dus geen toetsbaar begrip. Deze vaagheid van agile is een probleem voor het maken van afspraken over agile werken.



Vrijwel alle organisaties lossen dit op door de keuze van een specifieke methode die de agile bedoelingen invult. In Nederland is dit vrijwel altijd Scrum, uitgevonden door Ken Schwaber en Jeff Sutherland. Agile coach Rini van Solingen van Prowareness heeft zijn boeken bijvoorbeeld “De kracht van scrum” en “Scrum voor managers” genoemd, en ook Michael Franken van Zilverline schreef “Scrum voor dummies” om de agile

aanpak uit te leggen. Het is aan te raden om ook in contracten dit voorbeeld te volgen en het woord agile overal door scrum te vervangen. Je legt vast om de methode scrum te gebruiken, traint mensen in scrum, en vult de scrum rollen en bijeenkomsten in. Afwijken van scrum kan altijd, maar kan en beste doen door eerst voor scrum te kiezen en dan een aantal aanvullende of uitzonderende regels toe te voegen. Op die manier kiest men welk een duidelijk gedefinieerde aanpak.



Figuur 1: Illustratie van het scrum-proces.

Bron: Lakeworks, via wikipedia

De methode scrum is gedefinieerd in een “scrum guide”. Deze beschrijft de rollen, meetings en planningshulpmiddelen van de methode: zo moet er onder scrum bijvoorbeeld een product backlog zijn met een beschrijving van de nog te ontwikkelen functies, en moet er een planning sessie, een sprint review en een retrospective worden gepland, en moet er een scrum

master benoemd worden die het proces bewaakt en een product owner die de richting kiest. Er worden ook door meerdere Nederlandse partijen trainingen aangeboden voor zowel de IT-ers en de betrokken medewerkers van

de opdrachtgever.

Vraag 2: Wordt agile in de praktijk gebruikt?

Zoals in de inleiding al opgemerkt is er sprake van een agile-paradox. Volgens de IT-ers is 80% van de projecten al agile, volgens de juristen is 80% van de projecten niet agile. De oplossing van dit raadsel bestaat uit de combinatie van vier effecten:

- Veel organisaties die bewust kiezen voor agile, zien IT als strategisch en laten de projecten uitvoeren door interne medewerkers. Voor deze projecten wordt geen contract opgesteld en deze projecten zijn daardoor niet zichtbaar voor de juristen.
- Veel ontwikkelteams kiezen voor scrum als werkwijze, maar communiceren dit niet expliciet naar de klant. In plaats van ‘scrum’ expliciet in het contract af te dwingen, maakt men op

lager niveau werkafspraken met vertegenwoordigers van de opdrachtgever over frequent contact en tussentijds bijstellen.

- Veel projecten gebruiken elementen uit agile of scrum, maar niet de gehele methodiek. Aanvankelijk van of men het glas halfvol of halfleeg acht, kan men het wel of niet als een agile project beschouwen. Typisch gebruikt men wel het intensieve contact, maar maakt men wel afspraken over een vast eindresultaat voor een vaste prijs. IT-ers noemen deze projecten vaak agile omdat voor het contact het belangrijkste is. Juristen concentreren zich vaak op de al dan niet aanwezige resultaatverplichting en noemen het geen agile-project.
- Veel IT-advocaten zijn gespecialiseerd in grotere, complexe projecten waarin ook hardware-ontwikkeling en organisatieverandering opgenomen is. Denk aan de eerste invoering van een ERP-systeem. Binnen deze context is agile (en scrum) nog steeds bruikbaar, maar is het niet verstandig om de totale aanpak als 'scrum' of agile te omschrijven. Men kan beter concrete fases en mijlpalen benoemen, waardoor het contract een waterval-contract lijkt. Binnen deze waterval-projecten zullen er nog steeds teams zijn die via scrum werken.

Chargerend kan men zeggen dat één derde van de IT-ontwikkeling in Nederland volledig agile gedaan wordt, één derde compleet niet-agile en één derde zich in een grijs gebied bevindt. Op zich is dit grijze gebied geen probleem. Eén van de agile principes is immers om zich niet blind te staren op de contractuele afspraken: voor een goed eindresultaat is een goede samenwerking belangrijk. Het contract heeft op slechts enkele belangrijke momenten een functie: bij acceptatie en afronding, bij geschillen die niet op laag niveau oplosbaar bleken, en bij uit elkaar gaan van partijen.

Zoals één van de ge-interviewden opmerkte: “Collega’s van mij waren met een IT-contract bezig waarin software iteratief ontwikkeld werd en regelmatig gedemonstreerd. Op een gegeven vroeg ik of er agile werd gewerkt. Dit bleek inderdaad het geval, ook al werd dat niet zo benoemd”.

Vraag 3: Wordt agile aangeraden vanuit juridisch perspectief?

Puur technisch gezien heeft agile werken volgens scrum duidelijke voordelen. Niet alleen volgens de agile-adepten maar ook volgens de juristen: “door regelmatig werkende software te tonen wordt teleurstelling aan het einde voorkomen”. Agile werken zorgt er ook voor dat een tegenvallende resultaat minder vaak tot een juridisch geschil leidt: de klant is eerder en beter op de hoogte van deze resultaten en heeft de mogelijkheid de ontwikkeling stop te zetten.

Tegelijkertijd is agile geen wondermiddel: de kans op tegenslagen bij de ontwikkeling wordt als even groot ingeschat. Veel van de problemen in IT-projecten worden niet veroorzaakt of beïnvloed door de ontwikkelmethode. De meeste van de geïnterviewden staan neutraal tegenover agile of zouden agile aanraden, zij het onder voorwaarden: als het project zich ervoor leent.

De volgende risico's ontstaan wel of zijn we extra groot bij een agile aanpak:

- **Architectuur:** Voor bepaalde eisen (schaalbaarheid, betrouwbaarheid, beveiliging) is het noodzakelijk dat er van te voren al een opzet gekozen is die dit mogelijk maakt. Bij een agile aanpak kan het gebeuren dat men hier geen aandacht aan besteedt en op de verkeerde manier begint.
- **Requirements:** Agile kan een excuus zijn om te beginnen met de bouw zonder dat de alle requirements verzameld zijn, en dus ook zonder dat een schatting van de totale kosten gemaakt kan worden. Als later een schatting tegenvalt geeft dat een lastige situatie.
- **Afwijken van het eindresultaat:** Onderdeel van agile is om te werken in korte sprints, en per sprint te kiezen welke features worden gerealiseerd. Als de vertegenwoordiger van de opdrachtgever het einddoel niet voor ogen houdt maar inzoomt op details, wordt een perfecte voordeur afgeleverd maar de rest van het huis vergeten.
- **Geen aansprakelijkheid leverancier bij mislukken project:** Dit is een heikel punt omdat de belangen van de leverancier en opdrachtgever hier lijnrecht tegenover elkaar staan. Uitgangspunt zou moeten zijn dat de risico's verdeeld worden op een manier dat beide partijen de risico's kunnen dragen. Met name als het project groot is voor de opdrachtgever, en uitgevoerd wordt door een grote partij als leverancier is het feit dat de opdrachtgever alle risico's neemt een probleem.

Al deze risico's zijn te vermijden door je als opdrachtgever in de methode te verdiepen, en vervolgens ook intensief betrokken te zijn bij het project. De opdrachtgever moet zich eigenaar voelen van de architectuur en requirement, en zelf pas akkoord geven voor de start van agile ontwikkelen als hij/zij een duidelijk beeld heeft van de wensen. Zonder betrokkenheid van de opdrachtgever mislukt een agile project.

Eén van de advocaten zou een volledig agile contract voor zijn cliënten afraden: *“De IT-projecten die ik begeleid zijn vaak zo groot dat de opdrachtgever zich geen mislukte start kan permitteren, en het risico niet alleen kan dragen. De leverancier moet dus resultaatverantwoordelijk gemaakt worden, en dus moeten er specificaties worden opgesteld waar de leverancier zich aan moet houden. Bepaalde agile elementen kunnen wel, maar het blijft een klassiek ontwikkelcontract”*. Andere advocaten zouden de keuze voor wel of niet agile van het project laten hangen, of agile zelfs aanraden: *“De kans op complicaties is met agile hetzelfde, maar met agile is de kans op vroege ontdekking groter en kunnen grote schades en daarmee conflicten worden vermeden”*. Ook meerdere keren vermeld is dat agile werken projecten kleiner maakt, waardoor de risico's afnemen.

Vraag 4: Hoe maak je een goed IT-contract?

De voorwaarden voor een goed IT-contract voor een agile ontwikkelproject voor een groot deel overlappen met de eisen voor een normaal contract. Uiteindelijk gaat het erom dat de volgende zaken duidelijk worden vastgelegd:

- Scope: wat wordt er ontwikkeld?
- Kwaliteit: aan welke normen moet het eindresultaat voldoen?
- Tijd: welke mijlpalen moeten wanneer worden bereikt?
- Kosten: Hoeveel en wanneer wordt er betaald?
- Proces: Hoe wordt er gewerkt en vooral ook gestuurd?

Het contract moet op al deze punten zo precies mogelijk weergeven wat er is afgesproken, om misverstanden te vermijden.

De volgende aanbevelingen helpen om dit goed te doen:

- Maak een kort en leesbaar contract, dat voor ook voor niet-juristen begrijpelijk is. Liever twee tot vijf dan twintig tot vijftig pagina's
- Neem belangrijke inhoudelijke stukken zoals projectplannen, use cases, product backlogs en ontwerpen expliciet op als bijlage, zodat later duidelijk is welke versie bedoeld wordt.
- Organiseer als jurist eventueel een uitlegssessie na sluiten van het contract om degenen die het werk echt moeten uitvoeren uit te leggen wat afgesproken is
- Maak concrete afspraken over het proces tijdens het project. Spreek bijvoorbeeld af hoe vaak en wat voor overleg er is, wie er besluiten neemt over wijzigingen, en wie van de opdrachtgever op de hoogte gehouden wordt.
- Zorg ervoor dat betalingen gekoppeld zijn aan mijlpalen, zodat er alleen betaald wordt als er daadwerkelijk iets is geleverd.
- Wees als opdrachtgever betrokken bij de voortgang, en bezoek het ontwikkelteam minimaal om de week.
- Spreek af hoe de leverancier kan escaleren als er niet voldoende betrokkenheid van de opdrachtgever is.
- Doe aan contract-onderhoud: controleer geregeld welke aanvullende afspraken gemaakt zijn of gemaakt moeten worden en voeg deze toe aan het contract. Spreek bijvoorbeeld af om elke 3 maanden het contract bij te stellen.
- Denk na over beëindigings-scenario's van de overeenkomst. Hoe kan men uit elkaar gaan als één van de partijen niet tevreden is of er onverwachte ontwikkelingen zijn? Zorg ervoor dat het contract duidelijk beschrijft wanneer en hoe men uit elkaar gaat.

Vraag 5: Wat zijn specifieke agile contractafspraken?

Deze tips gelden zowel voor projecten met en zonder agile aanpak. Daarnaast zijn er enkele specifieke tips voor agile:

- Leg een concrete methode vast. Gebruikelijk in Nederland is om te kiezen voor scrum.

- Leg expliciet vast wie de scrumrollen inneemt (scrum master en product owner) en welke ‘stakeholders’ bij de demonstratie moeten zijn.
- Zorg aan de opdrachtgevende kant voor de best mogelijke ‘product owner’: iemand met mandaat die drie tot vijf dagen per week vrijgemaakt is en minimaal drie keer per week naar het ontwikkelteam gaat om de voortgang te bekijken en feedback te geven op de reeds gerealiseerde elementen.
- Zorg voor opleiding. Zowel voor de mensen ingezet door de leverancier en voor de betrokkenen aan de kant van de opdrachtgever. Er zijn goede en goedkope scrum-opleidingen van 1 of 2 dagen die ook op locatie kunnen worden gegeven.
- Leg bij een vaste-prijs contract goed vast of en door wie de scope van het project mag worden aangepast. In principe mag bij scrum de product owner de scope wijzigen op grond van nieuwe inzichten. Wie als opdrachtgever wil dat er zo min mogelijk verandert, zal voortdurend moeten communiceren met de product owner (of de rol zelf moeten invullen).
- Zorg ervoor dat er binnen de opdrachtgevende organisatie goed gecommuniceerd wordt over de doelen van het project en de eisen aan het systeem. Het is met name bij agile gevaarlijk als betrokkenen van de opdrachtgever niet de juiste kant op sturen.
- Overleg aan het begin van het project ook over niet-functionele eisen, zoals aantallen gebruikers, snelheid en beveiliging. Bespreek niet alleen wat de eisen zijn maar ook hoe deze gedemonstreerd kunnen worden tijdens het project.

Met dank aan

Dit artikel is tot stand gekomen met dank aan de volgende advocatenkantoren:

- Allen & Overy
- Baker & McKenzie
- Brinkhof advocaten
- De Gier Stam & Advocaten
- ICTrecht
- Legaltree
- Leopold Meijnen Oosterbaan
- Louwers Advocaten
- Project Moore
- SOLV advocaten
- Seelt IT & Commercial contracting
- Ventoux
- Vondst advocaten
- Wisemen advocaten

Bronnen

- Agile manifesto: <http://agilemanifesto.org>
- Scrum guide, beschikbaar via www.scrum.org. De Nederlandse vertaling is hier beschikbaar: <https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide-NL.pdf>
- De kracht van scrum, boek door Rini van Solingen. Pearson 2010. <http://rinivansolingen.nl/boeken/de-kracht-van-scrum/>
- Whitepaper Aandachtspunten bij TI-contracten, door Rutger Ketting, uitgegeven door Nysingh advocaten
- ICT~Office voorwaarden Module 2 ontwikkeling van programmatuur, uitgegeven door Nederland ICT (voorheen ICT~Office)
- Agile contracts primer, derived from the book Practices for Scaling Lean & Agile Development: Large, Multisite, & Offshore Product Development with Large-Scale Scrum by Tom Arbogast, Craig Larman, and Bas Vodde

- Eisen aan IT-contracten , artikel in Informatie 2010-6 door Polo van der Putt en Eva de Vries
- Bird&Bird – contracting for agile development position paper September 2012
<https://www.twobirds.com/~media/PDFs/Brochures/Contracting%20for%20Agile%20software%20development%20projects.pdf>
- Software: Deskundig en praktisch juridisch advies. Boek uitgegeven door ICTrecht, te vinden op <https://ictrecht.nl/boeken/software/>

Bron titel-afbeelding: PierreSelim – creative commons