

# Keuze ontwikkelmethode

**Hoe kom ik tot de best passende ontwikkelmethode voor mijn project(en)?**



# Auteurs



## Jan Klabbers

Jan Klabbers is een ervaren requirementsspecialist en softwareontwikkelproces-engineer bij Capgemini, met de meeste ervaring in maatwerksoftwareontwikkeling. Hij is op dit gebied actief voor de overheid, het openbaar vervoer, de gezondheidszorg, productiebedrijven en verzekeraars. Binnen Capgemini is hij trekker van de requirements-specialisten community om te komen tot een verder gestandaardiseerde werkwijze.



## Erik Hooft

Erik Hooft is projectmanager bij Capgemini met meer dan tien jaar ervaring in projectmanagement en meer dan vijftien jaar in maatwerksoftwareontwikkeling. Hij heeft vele projecten uitgevoerd met verschillende ontwikkelaanpakken, zoals Waterval, RUP en Scrum. Hij is onder meer actief voor klanten in de financiële sector, de overheid en de private sector. Het werken met offshoreteams (zoals in India) is een van zijn specialiteiten en hij werkt er continu aan om de manier van werken binnen projecten te optimaliseren.

# Voorwoord

De directe aanleiding voor het schrijven van de eerste versie van deze whitepaper was een klantvraag: geef een beknopte richtlijn voor het kiezen van een goede ontwikkelmethode. En natuurlijk liever morgen dan volgende week.

Dit artikel is daarmee vooral de weerslag van de praktijkervaring van twee Capgemini-consultants met ruime ervaring in maatwerk softwareprojecten. Deze mening is getoetst door collega's waardoor nog meer praktijkervaring en theoretische kennis is ingebracht in de huidige versie. En gezien de reacties is het wel een onderwerp dat leeft.

Dit document is de weerslag van discussie, die ons een handleiding geeft in de beslissing welke ontwikkelmethode te hanteren op een project.

Een belangrijke observatie die aan deze whitepaper ten grondslag ligt, is dat bedrijven vaak al te gemakkelijk één ontwikkelmethode omarmen en deze vervolgens geforceerd op al hun projecten willen toepassen. Vaak met de nodige problemen en weerstand in de organisatie. Ons uitgangspunt - gevoed door Capgemini's Technovision 2011 - is dat elke ontwikkelmethode zijn sterke en zwakke punten heeft en dat er factoren zijn die een methode meer of minder geschikt maken voor een project.

Het inschatten van de genoemde criteria vergt de nodige praktijkkennis. Pas bij doorvragen op bepaalde onderwerpen zal de benodigde informatie boven tafel komen. Deze whitepaper en bijbehorende tool<sup>1</sup> is de basis waarop deskundigen gezamenlijk met de klant kunnen onderzoeken welke methode het beste past.

Utrecht, juli 2012

---

<sup>1</sup> De grafieken die in dit document getoond worden, zijn afkomstig uit de genoemde tool.

# Inhoud

---

1 Managementsamenvatting	3
2 Inleiding	5
3 Ontwikkelmethoden	7
4 Aspecten en criteria	9
4.1 Opdrachtgevareigenschappen	9
4.1.1 Beschikbaarheid van stakeholders	9
4.1.2 Cultuur van delegeren	9
4.1.3 Beslisvaardigheid	10
4.1.4 Stabiliteit van scope en requirements	10
4.1.5 Omgaan met onzekerheid	10
4.1.6 Mogelijke deploymentfrequentie	11
4.1.7 Businessvolwassenheid in ICT	11
4.2 Opdrachtnemereigenschappen	11
4.2.1 Senioriteit van het team	11
4.2.2 Medewerkers multidisciplinair	12
4.3 Contract	12
4.3.1 Verwachting ten aanzien van resultaat	12
4.3.2 Afrekenmethoden	12
4.3.3 Afspraken over oplevermoment	12
4.3.4 Afspraken over functionele scope in het contract	13
4.4 Productaspecten	13
4.4.1 Soort toegevoegde waarde	13
4.4.2 Bekendheid technologie	13
4.4.3 Time-to-market	13
4.4.4 Releasecyclus	14

# 1 Management-samenvatting

Het kiezen van een ontwikkelmethode passend bij opdrachtgever, opdrachtnemer, contractvorm en het type product dat gemaakt moet worden, is een complexe keuze. Op basis van ervaringen uit de praktijk geeft deze whitepaper een handleiding die helpt bij het maken van een keuze.

Hierbij wordt niet op alle ontwikkelmethoden ingegaan, maar worden ze opgedeeld in drie hoofdgroepen met elk hun methodische kenmerken:

- **Lineair** - geen herhaling van processtappen
- **Iteratief** - vooraf gedefinieerde set van herhalingen
- **Agile** - vooraf gelimiteerde set van herhalingen

Hybride vormen zijn ook mogelijk; wij stellen ons een schaal voor met links lineair, in het midden iteratief en rechts agile. Elk project is ergens op deze schaal te plotten.



Voor een meer uitgebreide beschrijving zie hoofdstuk 3 Ontwikkelmethoden.

De keuze wordt gemaakt door een aantal aspecten van opdrachtgever, opdrachtnemer, contractvorm en type product in overweging te nemen. Afhankelijk van de eigenschappen van het aspect, wordt een advies gegeven over de toepasbaarheid van elke groep ontwikkelmethoden:

- A. Is een voorwaarde voor een succesvol project.
- B. Zal de kans op een succesvol project verhogen.
- C. Het heeft geen invloed op het succes van het project.
- D. Beperkte maatregelen zijn noodzakelijk om de kans op een succesvol project op peil te houden.
- E. Uitgebreide maatregelen zijn noodzakelijk om de kans op een succesvol project op peil te houden.

Voor deze managementsamenvatting beperken we ons tot een voorbeeld:

Stel dat de beschikbaarheid van de medewerkers van de opdrachtgever, die direct aan het project deelnemen, laag is. Denk hierbij aan de stakeholders, of in een Scrum-project de product owner. Zijn deze beperkt aanwezig, dan is de score:

## Lineair - D

Beperkte maatregelen zijn noodzakelijk om de kans op een succesvol project op peil te houden.

**Iteratief - D**

Beperkte maatregelen zijn noodzakelijk om de kans op een succesvol project op peil te houden.

**Agile - E**

Uitgebreide maatregelen zijn noodzakelijk om de kans op een succesvol project op peil te houden.

Bij een lage beschikbaarheid (dus de medewerkers zijn aangehaakt, maar kunnen slechts incidenteel tijd investeren) zul je dit risico bij een lineair en iteratief project moeten managen en wordt voor een agile aanpak aangeraden om bewust maatregelen te treffen (bijvoorbeeld een hybride vorm tussen agile en iteratief). Dit natuurlijk op basis van dit ene criterium. Vanzelfsprekend moeten ook alle andere criteria in ogenschouw worden genomen.

Een geschikte aanpak is dus een aanpak die veel A, B en C scoort, beperkt D en het liefst geen E. De eigenschappen waarbij een D en E gescoord is, moeten gemanaged worden als een risico. Hierbij zullen de E-scores vaak in het risico-kwadrant grote impact, hoge kans van optreden liggen.

En de maatregelen die getroffen worden voor E-scores, zullen ervoor zorgen dat er van de gekozen methode afgewezen wordt met een hybride methode als resultaat.

## 2 Inleiding

In de wereld van maatwerksoftware is op dit moment een groot aantal ontwikkelmethoden beschikbaar die de leidraad vormen voor de ontwikkeling van maatwerksoftware (LAD, IAD, DSDM, RUP, Scrum, Smart etc.). De keuze die gemaakt wordt, is op dit moment afhankelijk van verschillende voorkeuren:

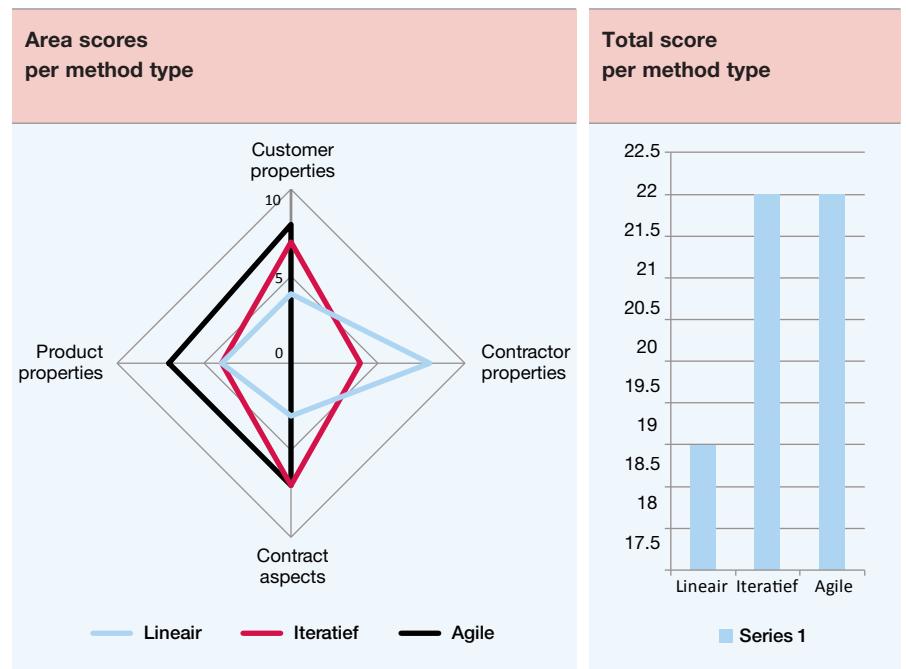
- voorkeur van de klant
- voorkeur van de ICT-leverancier
- voorkeur van de projectleider
- mode in de bedrijfstak

In dit document verzamelen we de best practices binnen Capgemini voor het maken van deze keuze, om zo tot een meer gefundeerde keuze te komen op basis van eigenschappen van:

- opdrachtgevende organisatie
- opdrachtnemer
- contractvorm
- het op te leveren product

Op basis van deze criteria hebben we een tool gemaakt die een grafische weergave geeft van de geschiktheid van elk van de genoemde ontwikkelmethodes.

Een voorbeeld van een willekeurig project:



Opbouw van dit document:

- In hoofdstuk 3 wordt eerst de definitie gegeven van de drie groepen van softwareontwikkelmethoden (lineair, iteratief en agile) die in dit document gehanteerd worden.
- Daarna worden in hoofdstuk 4 de criteria uitgewerkt voor het toepassen van een ontwikkelmethode.

# 3 Ontwikkelmethoden

Er bestaan veel verschillende ontwikkelmethoden in de markt. Wel zien we dat er grofweg twee of drie hoofdsoorten onderscheiden worden, maar ook dat deze drie niet hard gedefinieerd zijn.

Bij twee hoofdsoorten worden lineair (waterval) en iteratief onderscheiden.

Om bij drie te komen, wordt binnen iteratief een onderscheid gemaakt tussen methoden die toch zo veel mogelijk op voorhand specificeren en methoden waarbij het eindresultaat veel opener is. Voor de laatste methoden hanteren we in dit artikel de verzamelterm agile. Iteratief behouden we - bij gebrek aan een andere term - voor de methoden die meer op voorhand specificeren.

Het is voor ons belangrijk dit onderscheid tussen iteratief en agile te hanteren, omdat ze elk hun eigen factoren hebben die het succes van het project beïnvloeden.

Hybride vormen zijn ook mogelijk; wij stellen ons een schaal voor met links lineair, in het midden iteratief en rechts agile. Elk project is ergens op deze schaal te plotten. Een project dat opgestart wordt met een agile aanpak kan maatregelen moeten treffen om zich aan te passen aan de omstandigheden, waardoor het project een stuk naar links opschuift op de schaal.



Bijvoorbeeld: In een agile project blijkt dat de product owner<sup>2</sup> niet het beslismaat heeft, wat voor een strikt agile project nodig is (bijvoorbeeld doordat de cultuur van delegeren in het bedrijf dit onmogelijk maakt). Dan kun je een wekelijks stakeholderberaad inrichten, om de nodige beslissingen te nemen. Dit zal echter ten koste gaan van de agility van je project - het kan zijn dat je een week op een beslissing moet wachten. Hiermee schuift het project een stukje naar links op de schaal.

Hierna geven we de kenmerken van deze drie hoofdsoorten (lineair, iteratief en agile), zoals wij deze in dit document hanteren. We zijn ons ervan bewust dat we hiermee de termen meer specifiek maken, dan in de dagelijkse praktijk gemeengoed is.

<sup>2</sup> De product owner is in een Scrum-project de klantmedewerker met een mandaat vanuit de organisatie om te bepalen welke onderdelen van het product als eerste gerealiseerd worden. Hij zorgt ervoor dat het eindproduct de hoogst mogelijke toegevoegde waarde heeft.

Een lineaire ontwikkelmethode:

- kent geen herhaling van stappen in het ontwikkelproces;
- heeft een vast gedefinieerd eindproduct per fase (deliverable-gedreven proces);
- heeft gefixeerde requirements;
- documenteert het systeem vooraf;
- kent een document-gedreven kennisoverdracht;
- vraagt om hoge klantdeelname voorafgaand aan of in eerste fase van project;
- vraagt om beperkte klantdeelname tijdens het project na de eerste twee fasen, daarna wel betrokkenheid bij het reviewen en goedkeuren;
- besteedt beperkt aandacht aan het uitsluiten van technologierisico's;
- heeft een 'volgens specificatie' aanpak;
- heeft door manager aangestuurde teams;
- heeft teamleden met vaste rollen en verantwoordelijkheid voor 'eigen producten'.

Een iteratieve ontwikkelmethode:

- definieert in de eerste fase van het project de set iteraties die het project kent en de geadresseerde inhoud;
- evalueert in het gevolgde proces elke iteratie en wordt zo nodig aangepast;
- definieert vroeg in het traject 80% van de requirements;
- geeft de mogelijkheid van wijzigen van requirements na goedkeuring, middels een gecontroleerd proces van wijzigingen;
- documenteert het systeem tijdens het ontwikkelen;
- kent een document-gedreven kennisoverdracht, aangevuld met mondelinge sessies;
- vraagt om een geregelde klantdeelname tijdens het project;
- heeft het mitigeren van (technologie-)risico's als drijfveer;
- heeft een aanpak die een combinatie is van 'volgens specificatie' en 'waarde voor de business';
- kent door manager aangestuurde teams;
- heeft teamleden met vaste rollen en verantwoordelijkheid voor 'eigen producten'.

Een agile ontwikkelmethode:

- itereert in korte sprints van maximaal één maand; de inhoud van een sprint wordt bij de start van de sprint bepaald;
- geeft dagelijks de mogelijkheid de gevolgde werkwijze aan te passen;
- werkt met een geordende lijst van product features (product backlog) die gedurende het gehele project wordt bijgewerkt op basis van behoefte van klant/markt;
- geeft hoogste prioriteit aan het opleveren van op dat moment hoogste toevoegde waarde;
- documenteert het systeem in de mate die nodig is;
- legt nadruk op samenwerking en mondelinge kennisoverdracht;
- heeft een 'waarde voor de business' aanpak;
- kent multidisciplinaire teamleden, verantwoordelijk voor de opgeleverde producten;
- heeft een zelfsturend team;
- vraagt om een hoge deelname van de klant (in persoon van product owner).

We kiezen er bewust voor om de technieken die binnen een methode gebruikt worden (zoals informatieanalyse, technisch ontwerp, Use Cases, Smart Use Cases, UML, softwaregeneratie, wire frames, prototyping etc.) geen onderdeel van deze definitie te maken. In onze opvatting passen bepaalde technieken beter bij een methode en zijn misschien binnen deze methode uitgewerkt. Echter, er is geen beperking deze technieken ook binnen een andere methode toe te passen. Zo is bijvoorbeeld het gebruik van 'daily stand-ups' binnen meer agile methoden ontstaan en is daar een noodzakelijk onderdeel, maar het kan wel degelijk van toegevoegde waarde zijn binnen een lineair ontwikkeltraject. Maar het doen van 'daily stand-ups' betekent niet dat je agile werkt.

# 4 Aspecten en criteria

Voor de keuze van ontwikkelmethoden speelt een aantal aspecten een rol. Dit zijn eigenschappen van opdrachtgever, opdrachtnemer, contractvorm en het op te leveren product. Een eigenschap zal het succes van een project op verschillende manieren bepalen:

- A. Is een voorwaarde voor een succesvol project.
- B. Zal de kans op een succesvol project verhogen.
- C. Het heeft geen invloed op het succes van het project.
- D. Beperkte maatregelen zijn noodzakelijk om de kans op een succesvol project op peil te houden.
- E. Uitgebreide maatregelen zijn noodzakelijk om de kans op een succesvol project op peil te houden.

Een geschikte aanpak is dus een aanpak die veel A, B en C scoort, beperkt D en het liefst geen E. Een E beschouwen we echter niet als een showstopper maar als een signaal waarop maatregelen moeten worden getroffen. De eigenschappen waarbij een D en E gescoord is, moeten gemanaged worden als een risico. Hierbij zullen de E-scores vaak in het risicotkwadrant grote impact, hoge kans van optreden liggen.

En de maatregelen die getroffen worden voor E-scores zullen ervoor zorgen dat er van de gekozen methode afgeweken wordt met een hybride methode als resultaat.

## 4.1 Opdrachtgevereigenschappen

### 4.1.1 Beschikbaarheid van stakeholders

De verschillende ontwikkelmethoden vragen om een verschillende deelname van medewerkers van de opdrachtgever. Bij een lineair project zal dit bij momenten kort en intensief zijn. Puur agile projecten vragen om een constante deelname van de product owner.

Indien dit niet mogelijk is binnen de organisatie, dan kan overwogen worden om meer iteratief te werken waarbij wel de agile technieken optimaal gebruikt worden. De voordelen van agile worden in dat geval niet optimaal benut.

Beschikbaarheid	Lineair	Iteratief	Agile
Laag - maandelijkse enige dagen	D	D	E
Gemiddeld - wekelijks een dag	D	B	D
Hoog - dagelijks uren	C	B	A

### 4.1.2 Cultuur van delegeren

Als een organisatie geen cultuur heeft van het delegeren van beslissingsbevoegdheden en het mandaat bij het hoger management ligt, dan wordt het moeilijk een proces te voeren dat dagelijks om beslissingen vraagt. Hiermee bedoelen we ook organisaties waarbij formeel het mandaat laag in de organisatie ligt, maar waar informeel toch door het hoger management de uiteindelijke beslissing wordt genomen.

Een maatregel voor de methoden die op dit punt een E scoren kan zijn dat de managementlaag die uiteindelijk de beslissing neemt toch eens per week beschikbaar is. Hiervoor kan een wekelijkse vaste meeting ingepland worden. Ook dit is weer een maatregel die de agility van het project zal beperken en het project methodisch laat opschuiven richting een iteratieve aanpak.

Beslissingslaag	Lineair	Iteratief	Agile
Hoger management	B	E	E
Middel management	C	B	E
Gemandateerd aan opdrachtgever projectleden	C	B	A

#### 4.1.3 Beslisvaardigheid

Dit aspect ligt dicht bij het vorige, maar heeft een andere inslag. Heeft de organisatie een cultuur van snel beslissen of moet elke beslissing organisatiebreed afgestemd worden?

Binnen een agile aanpak zal het bij lagere beslisvaardigheid moeilijk worden om het benodigde mandaat uit te voeren. De kans is groot dat besluiten - als ze al worden genomen - teruggedraaid moeten worden.

Een maatregel zou in dat geval het organiseren van een wekelijkse bijeenkomst van stakeholders zijn, waar de benodigde beslissingen worden genomen. Ook dit is weer een maatregel die de agility van het project zal beperken en het project methodisch laat opschuiven richting een iteratieve aanpak.

Beslisvaardigheid	Lineair	Iteratief	Agile
Laag - organisatiebreed afstemmen	B	D	E
Gemiddeld - afdelingsniveau	C	B	E
Hoog - medewerkers nemen beslissingen	C	B	A

#### 4.1.4 Stabiliteit van scope en requirements

Weet de business voorafgaand aan het project al goed wat hij nodig heeft en welke eisen aan het systeem gesteld worden?

Stabiliteit	Lineair	Iteratief	Agile
Hoog - 100%	A	B	C
Gemiddeld - 80% bekend	D	B	C
Laag - eindresultaat ligt open	E	D	B

#### 4.1.5 Omgaan met onzekerheid

Is een organisatie in staat om enige mate van onzekerheid over de uitkomst van het project te accepteren? Het is de natuur van de mens om van tevoren precies te weten hoe het systeem eruit komt te zien. Velen zien daarom de oplossing in lineair ontwikkelen. Dit ondanks het feit dat er ook tijdens lineair ontwikkelen onvermijdelijk behoeft is om wijzigingen door te voeren. Hierdoor is de uiteindelijke uitkomst niet meer volledig in lijn met het initiële ontwerp. Of worden wijzigingen tot nader order uitgesteld, waardoor een product niet meer aan de eisen van de organisatie voldoet.

Als een organisatie initieel niet kan omgaan met de ‘onzekerheid’ in een agile project, dan moet daar op voorhand aandacht aan worden besteed. Een goed uitgewerkte visie die de bandbreedte van het wijzigingsspeelveld inperkt, helpt om deze onzekerheid weg te nemen.

Omgang met onzekerheid	Lineair	Iteratief	Agile
Gaan voor 100% zekerheid	B	D	E
Kunnen leven met 80%	C	B	E
Minder mag ook	D	B	A

#### 4.1.6 Mogelijke deploymentfrequentie

Met welke frequentie kan de organisatie nieuwe versies van het product uitrollen op de verschillende omgevingen? Test- en acceptatieomgeving zijn hierbij cruciaal. Voor de productieomgeving kan dit ook gelden, als de wens is het product na iedere iteratie in productie te brengen.

Een maatregel zou kunnen zijn dat het projectteam omgevingen inricht speciaal voor het project. Ook hier geldt weer voor een agile project: kun je niet naar een gewenste omgeving deployen omdat de organisatie dat niet aankan, dan schuift het project sterk naar het midden van de schaal lineair-iteratief-agile op.

Als het ook geldt voor de testomgeving, dan gaat het project zelfs voorbij het midden, want een iteratieve aanpak verwacht ook in elke iteratie te kunnen testen.

Deploymentfrequentie	Lineair	Iteratief	Agile
Laag - maximaal eens per kwartaal	B	B	E
Gemiddeld - maandelijks	C	B	B
Hoog - wekelijks of meer	C	C	A

#### 4.1.7 Businessvolwassenheid in ICT

In hoeverre is de business ervaren in het doen van ICT-projecten.

Ervarenheid in doen projecten	Lineair	Iteratief	Agile
Laag - hooguit enkele projecten	D	B	D
Gemiddeld - één of twee per jaar	B	B	D
Hoog - meerdere parallelle projecten per jaar	D	C	A

#### Opmerking:

Het doen van lineaire projecten wordt negatief beïnvloed als er weinig ervaring is met het doen van ICT-projecten. Een volledig ontwerp vooraf vraagt veel voorstellingsvermogen van een medewerker van de opdrachtgever om het concrete eindresultaat bij het ontwerp voor te stellen.

### 4.2 Opdrachtnemereigenschappen

#### 4.2.1 Senioriteit van het team

Een team is een mix van professionals van verschillende senioriteit. Dit is ook afhankelijk van de beschikbaarheid. Soms is het gemakkelijker een iets groter team, maar met meer junior medewerkers, samen te stellen dan een heel ervaren team.

Senioriteit	Lineair	Iteratief	Agile
Laag - relatief veel junioren en medioren	C	D	E
Gemiddeld - mix maar junioren onder (bege)leiding	C	B	C
Hoog - nauwelijks junioren	C	B	A

De teamgrootte is natuurlijk van invloed op deze eigenschap. Hoe groter het team, des te gemakkelijker junioren een zinvolle rol kunnen hebben.

#### 4.2.2 Medewerkers multidisciplinair

Zijn de teamleden ervaren in één discipline (projectmanagement, requirements, analyse & design, bouw, test), of zijn ze thuis in meerdere disciplines?

Multidisciplinaire teamleden verhogen de snelheid van handelen en omschakelen en maken iteratief en incrementeel werken gemakkelijker te plannen.

In een agile project is een multidisciplinair team essentieel. Een maatregel - indien het team niet multidisciplinair is - is de samenstelling van het team opnieuw bepalen. Is dit niet mogelijk, dan zul je een meer iteratieve aanpak moeten kiezen en schuif je op de schaal op naar links.

Multidisciplinaire teamleden	Lineair	Iteratief	Agile
Weinig - 10%	B	D	E
Gemiddeld - 25% tot 50%	C	B	D
Hoog - > 75%	C	C	A

### 4.3 Contract

#### 4.3.1 Verwachting ten aanzien van resultaat

In de afspraken tussen opdrachtgever en opdrachtnemer worden afspraken gemaakt op welke wijze het resultaat beoordeeld wordt.

Resultaat	Lineair	Iteratief	Agile
Conform vooraf vastgestelde specificaties	A	C	D
Conform specificaties die gedurende het project gemanaged worden	D	B	D
Open binnen gestelde grote scope, tijd- en geldbeperking	E	D	B

#### 4.3.2 Afrekenmethoden

Een contract kent verschillende typen afrekenmethoden.

Afrekenmethoden	Lineair	Iteratief	Agile
Fixed - na aftekenen requirements	B	D	E
Fixed - na aftekenen requirements en risicoafdekking	B	B	E
Fixed - budget box	C	B	B
Time-material	B	B	C

Een maatregel voor agile zou in dit geval zijn om op basis van de voordelen die agile biedt een andere contractvorm met de klant te bespreken. Met het argument dat deze beter bij de gekozen methodiek passen en de toegevoegde waarde ervan zullen vergroten.

#### 4.3.3 Afspraken over oplevermoment

Op enig moment wordt er in een project een afspraak gemaakt tussen opdrachtgever en opdrachtnemer over het moment van opleveren.

Vaststellen oplevermoment	Lineair	Iteratief	Agile
Open	B	C	D
Na aftekenen requirements	B	D	D
Na aftekenen requirements en risicoafdekking	B	B	D
Time box	D	B	A

#### 4.3.4 Afspraken over functionele scope in het contract

De definitie van de functionele scope van een project kan in verschillende mate vastliggen in het contract. Dit lijkt op het aspect 4.1.4 Stabiliteit van scope en requirements. Er kan echter een groot verschil zijn tussen de manier waarop dit in het contract is vastgelegd en de werkelijke stabiliteit van de requirements.

Functionele scope	Lineair	Iteratief	Agile
Fixed	A	B	D
Gemiddeld - 80% bekend	D	B	D
Open	E	D	B

#### 4.4 Productaspecten

##### 4.4.1 Soort toegevoegde waarde

De toegevoegde waarde van een product kan verschillend zijn. De waarde kan liggen in het ondersteunen van basale processen en het voldoen aan wet- en regelgeving (compliancy). Heb je het niet, dan staat je business stil, maar je onderscheidt je er niet mee in de markt. Of een product heeft juist businesswaarde, het moet het bedrijf onderscheiden in de markt.

Toegevoegde waarde	Lineair	Iteratief	Agile
Compliancy	B	B	C
Mix compliancy en businesswaarde	D	B	C
Business waarde	D	D	B

##### 4.4.2 Bekendheid technologie

Het kan zijn dat vooraf bekend is dat een product met bewezen technologie gebouwd wordt, of dat er juist iets nieuws wordt uitgeprobeerd.

Technologie	Lineair	Iteratief	Agile
Bewezen	A	C	B
Nieuw - maar expliciet doel project	E	B	C
Nieuw - geen doel van project	E	B	D

Het werken met nieuwe technologie is in een agile aanpak mogelijk, indien het bewijzen van deze technologie het expliciete doel is van het project en businessdoelstellingen een secundaire rol spelen.

Dit zal dan een apart project zijn met een product owner speciaal voor het bewijzen van de nieuwe technologie. Het project om tot het uiteindelijke product met deze nieuwe technologie te komen, kan door een andere product owner worden uitgevoerd.

##### 4.4.3 Time-to-market

Voor sommige producten is het aanvaardbaar dat de weg naar implementatie lang is. Bij andere zal het heel kort moeten zijn, omdat anders de concurrentie de business wegkapt.

Time-to-market	Lineair	Iteratief	Agile
Lang - eerder een jaar	A	B	D
Gemiddeld - kwartaal	D	B	C
Hoog - korter dan maand	E	E	B

#### 4.4.4 Releasecyclus

Dit aspect ligt dicht tegen het vorige aan, maar er kunnen andere redenen zijn dan time-to-market, die de hartslag van de releasecyclus bepaalt.

Releasecyclus	Lineair	Iteratief	Agile
Jaar	A	B	C
Kwartaal	D	B	C
Maand of minder	E	E	B



## Over Capgemini

Met ruim 120.000 mensen in 40 landen is Capgemini wereldwijd een van de meest vooraanstaande aanbieders van consulting-, technology- en outsourcingdiensten. In 2011 rapporteerde Capgemini Group een omzet van 9,7 miljard euro. Samen met zijn klanten creëert en realiseert Capgemini resultaatgerichte business- en technology-oplossingen, toegesneden op de klant-

behoefte. Als een cultureel diverse organisatie heeft Capgemini zijn eigen onderscheidende manier van werken, de Collaborative Business Experience™. Hierbij maakt Capgemini gebruik van het wereldwijde leveringsmodel Rightshore®. Meer informatie via  
**[www.nl.capgemini.com](http://www.nl.capgemini.com)**

Rightshore® is een handelsmerk van Capgemini

### Capgemini Nederland B.V.

Papendorpseweg 100  
Postbus 2575 - 3500 GN Utrecht  
[jan.klabbers@capgemini.com](mailto:jan.klabbers@capgemini.com)  
Tel +31 30 689 84 54  
[erik.hooft@capgemini.com](mailto:erik.hooft@capgemini.com)  
Tel +31 30 689 53 44





**Capgemini Nederland B.V.**

Papendorpseweg 100  
Postbus 2575 - 3500 GN Utrecht  
Tel. 030 689 00 00