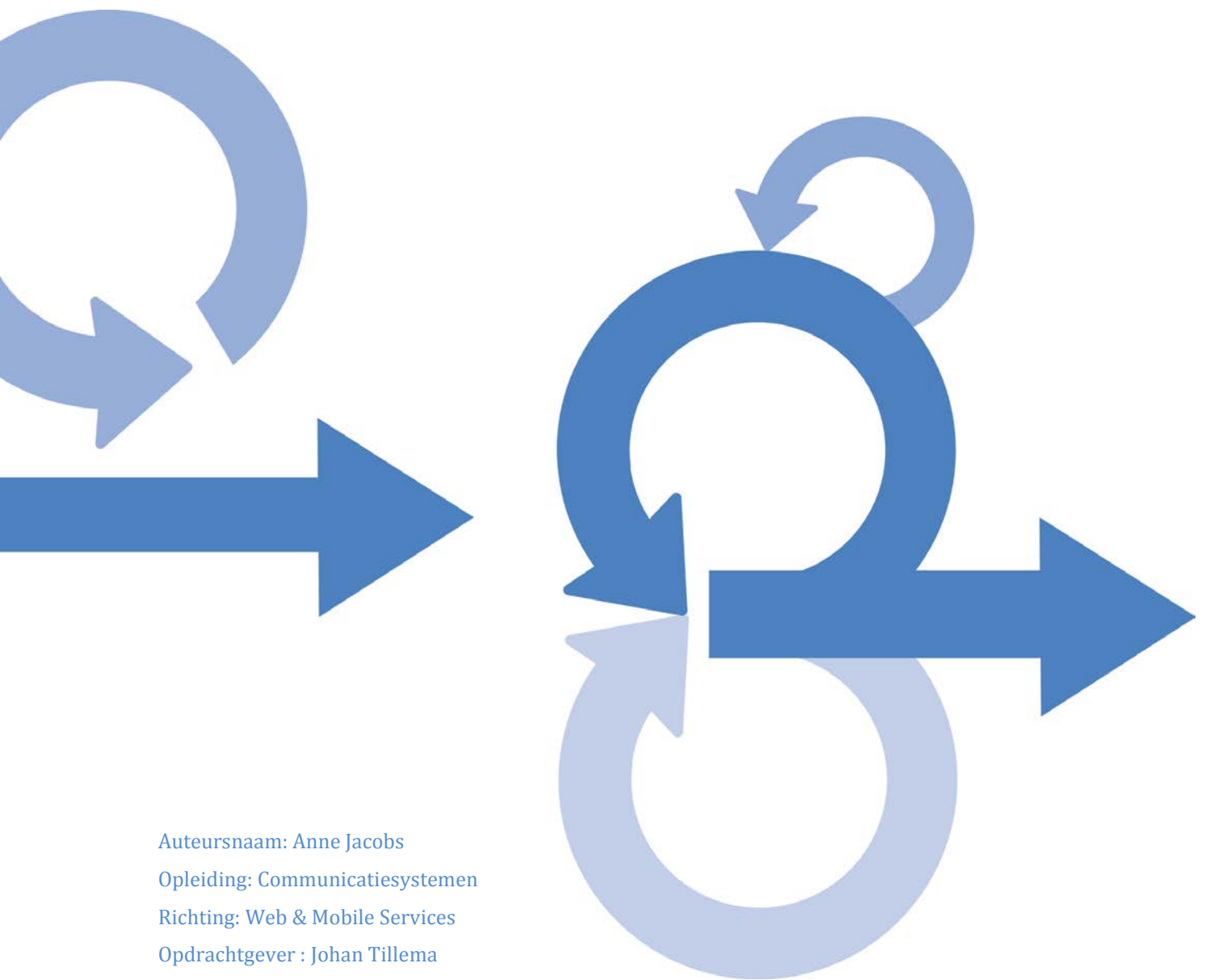


# Analyse en ontwerp in een Scrumproject



Auteursnaam: Anne Jacobs  
Opleiding: Communicatiesystemen  
Richting: Web & Mobile Services  
Opdrachtgever : Johan Tillema  
Datum: 09-01-2014



# Analyse en ontwerp in een Scrumproject

Auteursnaam: Anne Jacobs

Studentnummer:344753

Opleiding: Communicatiesystemen

Richting: Web & Mobile Services

Vakcode: XCMH1ASO

Opdrachtgever : Johan Tillema

Afstudeerbegeleider: Peter-Jan Hagedoorn

Datum: 09-01-2014

Plaats: Groningen



# Samenvatting

## Achtergrond

Softwareontwikkelpromjecten worden vandaag de dag op grote schaal uitgevoerd. Hierdoor is er een behoefte ontstaan om deze projecten te beheersen met behulp van een ontwikkelmethode. Op dit moment heerst er grote populariteit rondom Agile-ontwikkelmethoden. Scrum is een voorbeeld van een Agile-ontwikkelmethode. Waar Agile een aantal richtlijnen voorschrijft, beschrijft Scrum concrete regels om deze toe te passen. Voorafgaand aan Scrum werd veelal de Watervalmethode toegepast. Deze manier van werken is echter terrein aan het verliezen. Dit komt hoofdzakelijk doordat deze methode het niet goed toelaat om in te spelen op wijzigende behoeften bij de klant/opdrachtgever. Scrum is daarentegen een iteratieve methode waarbij niet alles op voorhand wordt gespecificeerd. Hoewel het werken volgens de Scrummethode grote voordelen oplevert, beschrijft Scrum hoofdzakelijk de basis van softwareontwikkeling. Er worden vooral richtlijnen gegeven met betrekking tot de realisatie.

Deze thesis beschrijft een onderzoek dat is gedaan ter verbetering van het analyse- en ontwerpproces in Scrumprojecten. Dit is gedaan in opdracht van softwareontwikkelfbedrijf Quintor. Quintor werkt zelf al jaren volgens de Scrummethode en is op zoek naar een gestandaardiseerde aanpak op het gebied van analyse en ontwerp. Het uitvoeren van analyse en ontwerp is essentieel bij softwareontwikkelpromjecten. Het is belangrijk dat dit gebeurt op een wijze die binnen Agile/Scrum past zodat de voordelen die worden behaald blijven bestaan. Een dergelijke aanpak dient daarom gebaseerd te zijn op theorieën, ervaringen en praktijksituaties.

## Doelstelling

De doelstelling van dit onderzoek luidt als volgt:

Het doel van het onderzoek is aanbevelingen doen aan Quintor ter verbetering van het analyse- en ontwerpproces in Scrumprojecten

*door*

op basis van een vooronderzoek en vastgestelde theorieën (over: requirement engineering, ontwikkelmethodes, Agile en Scrum) te komen tot een referentiekader om dit vervolgens te confronteren met klanten van Quintor en medewerkers van Quintor die werkzaam zijn bij deze klanten. Op deze manier zal inzicht worden gegeven in de werkwijze die zij hanteren met betrekking tot analyse en ontwerp in Scrumprojecten. Hierbij zal aandacht zijn voor zowel succesfactoren als knelpunten die optreden in praktijksituaties. Een beoordeling van de analyseresultaten die voortvloeien uit dit onderzoek zal bijdragen aan de oplossing van het probleem.

## Centrale vragen

Op basis van de doelstelling is er een onderzoeksmodel opgesteld waaruit de volgende centrale vragen zijn afgeleid:

- *Centrale vraag 1: Hoe ziet het referentiekader eruit gelet op het vooronderzoek en de geraadpleegde theorieën?*
- *Centrale vraag 2: Hoe wordt in de praktijk een invulling gegeven aan analyse en ontwerp binnen Scrum gelet op het beschreven referentiekader?*
- *Centrale vraag 3: Wat leert ons de beoordeling van de analyseresultaten met het oog op het doen van aanbevelingen op het gebied van analyse en ontwerp in een Scrumproject?*

## Onderzoek

Het beantwoorden van de centrale vragen is gebeurd door het onderzoek op te delen in twee fasen. In de eerste fase is er een kwalitatief vooronderzoek bij Quintor uitgevoerd en is er deskresearch naar verschillende theorieën gedaan. Aan de hand van deze bevindingen is er een referentiekader opgesteld. Tijdens de tweede fase is er een kwalitatieve vergelijkende casestudy uitgevoerd. In deze fase zijn er diepte-interviews afgenomen bij respondenten die bestonden uit klanten van Quintor en medewerkers van Quintor. De resultaten uit de vergelijkende casestudy zijn verwerkt door de interviews uit te werken in transcripten. Deze transcripten zijn vervolgens gecodeerd en gecategoriseerd. Vervolgens zijn de bevindingen beschreven en is er een conclusie geformuleerd. Deze zijn beschreven aan de hand van de drie hoofdthema's van deze thesis namelijk:

- het komen tot de Product Backlog;
- het ready maken van User Stories;
- de rol van de informatieanalist.

## Bevindingen en conclusie

Uit de bevindingen en de conclusie kan een aantal punten worden samengevat:

- Met betrekking tot het komen tot de Product Backlog blijkt dat het creëren van een bepaalde context essentieel is. Dit kan worden gedaan door met stakeholders de processen te doorlopen waarvoor de IT-ondersteuning nodig is. Door deze processen te doorlopen en benoemen kunnen User Stories worden geformuleerd. Duidelijk kwam de wens naar voren om deze context vast te leggen in een high level visueel artefact. De vorm hiervan kan verschillen zoals bijvoorbeeld een klikbaar prototype of een procesplaat met grafische modellen als Flowcharts. Dit artefact heeft naast dat het een hulpmiddel is tijdens het opstellen van de Product Backlog nog vele andere positieve werkingen. Zo geeft het gedurende het project zicht op de voortgang, creëert het een gemeenschappelijk taal en kan vertrouwen van stakeholders worden gewonnen.
- Het ready maken van User Stories is in de meeste gevallen een activiteit die parallel loopt aan de huidige sprint. Dit gebeurt bij grote projecten vaak door een losstaand team. Het bleek dat wat men verstaat onder 'ready' per project erg verschilt. Dit loopt uiteen van het formuleren van een

User Story tot het opstellen van een uitgebreide Use Case. Elementen gebruiken uit de Use Case en op het juiste moment op de juiste manier toepassen lijkt hiervoor de juiste middenweg. Het hanteren van de Definition of Ready gebeurt in veel gevallen niet of te weinig. Het blijkt noodzakelijk om dit wel te doen. Hetgeen dat opgeleverd moet worden moet voor iedereen duidelijk zijn in het team. Dit is voor het kunnen plannen van een sprint en commitment krijgen van het team cruciaal. Om de Definition of Ready te kunnen hanteren moet deze wel realistisch zijn en niet te hoge eisen bevatten.

- Met betrekking tot de rol van de informatieanalist blijkt dat deze bij de grotere projecten zich vaak begeeft in een losstaand team. In kleinere projecten is de informatieanalist vaak onderdeel van het ontwikkelteam. Het blijkt dat de rol van de informatieanalist het best past tussen het team en de Product Owner in. Wanneer er sprake is van een losstaand ready-team blijkt het belangrijk dat de informatieanalist wel betrokken blijft bij het behalen van het sprintdoel. Hiervoor moet de informatieanalist dus ook beschikbaar blijven tijdens de sprint voor eventuele specificatie.

Er blijken in de praktijk veel factoren van invloed op de werkwijzen op het gebied van analyse en ontwerp. Dit zijn onder andere: werken op gescheiden locaties, een bepaalde cultuur, kennis in de organisatie die nodig is bij het realiseren van de IT-ondersteuning, documentatie-eisen en informatiebehoefte vanuit het team. Hieruit bleek dat ieder project verschilt en om een andere aanpak vraagt. Hoewel deze thesis is ingestoken op analyse en ontwerp blijkt het vaak noodzakelijk om het gehele proces in kaart te brengen en verschillende lagen in de organisatie mee te nemen in het iteratieve ontwikkeltraject. Het recent ontwikkelde Scaled Agile Framework is hiervoor een goed uitgangspunt gebleken. Het Framework beschrijft een manier waarop je in diverse stadia op diverse niveaus binnen een organisatie om kunt gaan met analyse en ontwerp.

### **Aanbevelingen**

Naar aanleiding van het onderzoek is er een lijst opgesteld met richtlijnen op het gebied van analyse en ontwerp. De aanbeveling voor Quintor is om deze lijst toe te passen in Scrumprojecten. Tevens is er richting Quintor een aanbeveling gedaan om de binnen Quintor opgedane kennis en praktijkervaringen, die uit het vooronderzoek naar voren zijn gekomen, te gaan vastleggen en verspreiden om op deze manier ook andere organisaties te kunnen begeleiden in het analyse- en ontwerpproces.





# Voorwoord

Deze thesis is geschreven als laatste onderdeel van de opleiding Communicatiesystemen in de richting Web & Mobile services aan de Hanzehogeschool te Groningen. Met deze Bachelor thesis sluit ik de opleiding af. Gedurende de periode 1 september 2013 tot en met 9 januari 2014 heb ik onderzoek uitgevoerd in opdracht van Quintor. Quintor wilde met dit onderzoek op het gebied van analyse en ontwerp komen met een aanvulling op de softwareontwikkelmethode Scrum.

Het schrijven van de Bachelor thesis is een leerzame periode geweest die ook de nodige twijfels en onzekerheden met zich mee heeft gebracht. Uiteindelijk kijk ik zeer tevreden terug op de afgelopen periode. Ik wil een aantal mensen bedanken die mij erg hebben geholpen tijdens het schrijven van mijn thesis. Peter-Jan Hagedoorn voor zijn begeleiding en steun vanuit de Hanzehogeschool. Jan van Beek voor zijn begeleiding en steun vanuit Quintor. Twee begeleiders met veel inhoudelijke kennis en enthousiasme die mij de afgelopen periode geweldig hebben bijgestaan.

Ik wil Johan Tillema, directeur Quintor, bedanken. In eerste plaats voor het verstrekken van een opdracht die uitdagend en passend is geweest voor mij. Daarnaast voor de begeleiding tijdens het proces. Tot slot wil ik Jan Willem Omlo bedanken voor de extra begeleiding wanneer dit nodig was en alle mensen die ik heb mogen interviewen voor mijn casestudy.

Ik wil de lezer veel leesplezier wensen.

*Anne Jacobs*

*Groningen, december 2013*



# Inhoud

<b>Inleiding</b> .....	1
<b>Hoofdstuk 1 Onderzoeksopzet</b> .....	5
1.1 Organisatie .....	6
1.2 Projectkader .....	7
1.3 Doelstelling.....	10
1.4 Onderzoeksmodel .....	11
1.5 Centrale vragen .....	12
1.6 Methodologie & Strategie .....	13
1.7 Vooronderzoek.....	18
1.8 Theoretisch kader .....	19
<b>Hoofdstuk 2 Resultaten vooronderzoek en theoretisch kader</b> .....	20
2.1 Weergave resultaten vooronderzoek & theoretisch kader .....	21
2.1.1 Komen tot de Product Backlog.....	21
2.1.2 Ready maken.....	22
2.1.3 De rol van de informatieanalist.....	23
2.2 Het referentiekader .....	25
2.2.1 Komen tot de Product Backlog.....	25
2.2.2 Ready maken.....	25
2.2.3 De rol van de informatieanalist.....	26
<b>Hoofdstuk 3 Resultaten casestudy</b> .....	27
3.1 Weergave resultaten casestudy .....	28
3.1.1 Komen tot de Product Backlog.....	28
3.1.2 Ready maken.....	30
3.1.3 De rol van de informatieanalist.....	33
<b>Hoofdstuk 4 Conclusie</b> .....	34
4.1 Conclusie resultaten casestudy .....	35
4.1.1 Komen tot de Product Backlog.....	35
4.1.2 Ready maken.....	39
4.1.3 De rol van de informatieanalist.....	42

4.2 Advies in relatie tot de bevindingen .....	43
4.2.1 Komen tot de Product Backlog .....	43
4.2.2 Ready maken .....	43
4.2.3 De rol van de informatieanalist .....	44
4.3 Conclusie Scrumproces .....	45
<b>Hoofdstuk 5 Aanbevelingen</b> .....	<b>47</b>
5.1 Aanbevelingen Analyse en Ontwerp .....	48
5.2 Aanbevelingen vervolgonderzoek .....	49
<b>Hoofdstuk 6 Discussie</b> .....	<b>50</b>
6.1 Resultaten .....	51
6.2 Beperkingen .....	51
<b>Literatuurlijst</b> .....	<b>53</b>
<b>Bijlagen</b> .....	<b>55</b>
Bijlage 1: Topics en eventuele vragen .....	56
Bijlage 2: Samenvatting resultaten per case .....	58
Bijlage 3: Definition of Ready .....	62
Bijlage 4: Categorieën .....	64
Bijlage 5: Uiteindelijk categorieën per thema .....	65
Bijlage 6: Richtlijnen analyse en ontwerp in een Scrumproject .....	66





# Inleiding

Doordat softwareontwikkeling op grote schaal gebeurt, is ook de behoefte om softwareontwikkelprojecten te hanteren de afgelopen jaren sterk toegenomen. Tijdens softwareontwikkeling is het belangrijk om het proces zo uit te voeren dat er wordt voldaan aan de behoefte van de klant/opdrachtgever. Agile kan worden omschreven als een bepaalde gedachte over de uitvoering van softwareontwikkelprojecten. Meerdere ontwikkelmethodes proberen het gedachtegoed van Agile in praktijk te brengen. De populairste methode op dit gebied is Scrum. Scrum draait onder andere om het werken met zelfsturende multidisciplinaire teams, het inspelen op veranderingen en het werken in korte iteraties.

Voorafgaand aan Scrum stond onder andere de Watervalmethode. Het werken volgens de Watervalmethode heeft de afgelopen jaren verschillende problemen opgeleverd zoals het niet kunnen inspelen op een veranderende omgeving. Requirements worden bij de Watervalmethode in een vroeg stadium gedetailleerd vastgelegd. Er bleek dat wat er werd opgeleverd hierdoor vaak niet (meer) voldeed aan de behoefte van de klant.

Deze thesis is geschreven in opdracht van softwareontwikkelbedrijf Quintor. Quintor werkt al jaren volgens de Scrummethode en is op zoek naar een gestandaardiseerde aanpak op het gebied van analyse en ontwerp. Het doel van het onderzoek is aanbevelingen doen aan Quintor ter verbetering van het analyse- en ontwerpproces in Scrumprojecten

In dit onderzoek is er in eerste instantie een vooronderzoek bij Quintor uitgevoerd en is er vervolgens deskresearch gedaan naar relevante literatuur om op basis daarvan het referentiekader op te stellen. Vervolgens is er een kwalitatief onderzoek uitgevoerd in de vorm van een vergelijkende casestudy.

## **Opbouw**

In het eerste hoofdstuk van deze thesis wordt er een inleiding gegeven op de opdracht. Dit gebeurt in de vorm van een onderzoeksopzet. In dit hoofdstuk komen de organisatie, het projectkader, de doelstelling, het onderzoekmodel, de centrale vragen, de methodologie en de strategie aan bod. Het hoofdstuk wordt afgesloten met een toelichting op het vooronderzoek en het theoretisch kader. In hoofdstuk 2 worden de verkregen resultaten uit het vooronderzoek en theoretisch kader beschreven en wordt hieruit een referentiekader gevormd. In hoofdstuk 3 worden de verkregen resultaten vanuit de casestudy weergegeven en toegelicht. In hoofdstuk 4 wordt de conclusie in relatie tot de bevindingen vanuit de casestudy gegeven. Vervolgens wordt er in dit hoofdstuk beschreven wat de conclusie betekent voor het advies. Tevens wordt er een conclusie gegeven over het Scrumproces in relatie tot de bevindingen. In hoofdstuk 5 worden aanbevelingen gegeven die zijn gevormd vanuit de bevindingen en conclusie. De thesis wordt afgesloten met hoofdstuk 6 waarin een discussie is opgenomen. Hierin is aandacht voor de onderzoeksresultaten en beperkingen van het onderzoek.

Gedurende het hele rapport is ieder hoofdstuk zoveel mogelijk beschreven aan de hand van de drie hoofdthema's van deze thesis namelijk:

- het komen tot de Product Backlog;
- het ready maken van User Stories;
- de rol van de informatieanalist.







# Hoofdstuk 1

## Onderzoeksopzet

*Dit hoofdstuk kan worden gelezen als een inleiding op het onderzoek. Het hoofdstuk begint met het projectkader waarin is aangegeven wat de situatie is die de opdrachtgever als problematisch ervaart en waaraan de resultaten vanuit dit onderzoek een bijdrage leveren. (Verschuren & Dorenwaard, 2007) Het betreft achtergrondinformatie over de opdracht. Vervolgens wordt er informatie gegeven over de organisatie waarvoor het onderzoek is uitgevoerd. Hierna volgen de doelstelling, het onderzoeksmodel, de centrale vragen, de onderzoeksstrategie en de onderzoeksmethode. Deze geven weer hoe het onderzoek is uitgevoerd en welke vragen het beantwoordt. Het hoofdstuk wordt afgesloten met een toelichting op het vooronderzoek en theoretisch kader. Hier wordt verwezen naar de externe bijlage: Vooronderzoek & Theoretisch kader waar het daadwerkelijke resultaat is te vinden.*

## 1.1 Organisatie

Quintor is een organisatie die werkzaam is op het gebied van Agile software development, enterprise Java/.Net technologie en mobile development. Binnen Quintor is een breed scala van specialisten aanwezig op het gebied van softwareontwikkeling. Quintor richt zich hoofdzakelijk op het ontwikkelen via Java- en .Net-platformen. Naast softwareontwikkelaars bestaat het Quintorteam uit informatieanalisten en softwarearchitecten die gespecialiseerd zijn in analyse en ontwerp en het opstellen van een architectuur. Omdat mobile development de afgelopen jaren flink in opkomst is heeft Quintor zich naast software development ook in mobile development gespecialiseerd. Quintor ontwikkelt mobiele applicaties voor diverse platformen zoals Android en IOS.

Quintor is in 2005 opgericht en heeft momenteel 60 werknemers in dienst. Quintor heeft een vestiging in Groningen en in Amersfoort. Het uitvoeren van projecten gebeurt meestal in house bij de klant. Dit betekent dat wanneer een organisatie Quintor inhuurt, de werkzaamheden bij de klant worden uitgevoerd. Quintor neemt tijdens projecten de volledige realisatie op zich. Dit betekent dat ze de kennis kunnen leveren vanaf het begin tot het eind van projecten. Quintor pakt projecten volgens de Agile- / Scrummethode aan en heeft deze methode bij veel van haar klanten geïntroduceerd. Door gebruik te maken van deze methode bereikt Quintor een grote mate van betrokkenheid tussen stakeholders en het ontwikkelteam. Daarnaast leidt deze methode ertoe dat projecten sneller worden afgerond tegen minder hoge kosten.

Klanten zijn erg tevreden over het werken via de Scrummethode. Na afronding van een project blijven klanten veelal werken volgens de Scrummethodiek. Quintor is inmiddels specialist geworden in het invoeren van een Agile-denkwijze en Scrum in een organisatie. Organisaties worden door Quintor opgeleid en begeleid om te kunnen gaan werken via Agile/Scrum.

## 1.2 Projectkader

Quintor ontwikkelt software via de Agile-gedachte en de Scrummethode. Een ontwikkelmethode verschaft handvatten en richtlijnen die helpen bij het uitvoeren en hanteren van projecten. Scrum is een ontwikkelmethode die wordt gebruikt voor softwareontwikkeling. Scrum wordt meestal samen met de term Agile gebruikt. Agile kan worden omschreven als een bepaalde denkwijze (methodologie) die tijdens software development als uitgangspunt dient. Scrum is binnen deze methodologie een ontwikkelmethode die gebaseerd is op deze Agile-gedachte.

Scrum heeft als doel om te kunnen inspelen op veranderingen die plaatsvinden in projecten. Bij Scrum draait het om het werken in teams om een product te creëren. De teams werken in korte perioden, die sprints worden genoemd, aan een bepaald onderdeel van het product. De afgelopen jaren heeft de Agile-denkwijze en Scrum bij Quintor een grote rol ingenomen. Vrijwel alle projecten worden uitgevoerd via de Scrummethode.

Om dit projectkader te vormen is het vooronderzoek gebruikt dat te vinden is in de externe bijlage: *Vooronderzoek & Theoretisch Kader*. Hiervoor zijn de volgende mensen geïnterviewd:

Johan Tillema (opdrachtgever)	CEO Quintor
Jan van Beek (begeleider)	COO Quintor
René Boere	CTO Quintor
Jan Willem Omlo	informatieanalist Quintor

### Achtergrond

Voordat Scrum groot was in de wereld van softwareontwikkeling werd veelal de Watervalmethode gebruikt. De Watervalmethode gaat uit van verschillende fases waarin ieder onderdeel door iemand van een bepaalde expertise wordt uitgevoerd. Men heeft door de jaren heen geleerd dat deze manier van werken er vaak voor zorgt dat er problemen optreden. Dit ontstaat doordat alle eisen (requirements) in een vroeg stadium tot in detail worden vastgelegd. Vervolgens worden deze in een volgende fase gerealiseerd. Het probleem dat ontstaat bij deze aanpak is dat er vrijwel geen rekening wordt gehouden met een veranderende omgeving en veranderende behoefte bij de klant. Er is te vaak gebleken dat projecten die worden uitgevoerd op deze manier uitlopen en uiteindelijk niet opleveren waar de klant de meeste behoefte aan heeft. Dit ontstaat doordat klanten meestal pas tijdens de uitvoering van een project hun behoefte echt leren kennen. Hierdoor is de vraag naar een flexibele methode voor softwareontwikkeling gegroeid. De voordelen die met Scrum worden behaald is dat in tegenstelling tot de Watervalmethode, niet alles op voorhand tot in detail is vastgesteld, er continu contact is tussen klant en realisatieteam en er ruimte is om op veranderingen in te spelen.

Om een systeem te ontwikkelen is een inventarisatie van requirements waaraan het te ontwikkelen systeem moet voldoen noodzakelijk. Daarnaast dienen deze requirements te worden uitgewerkt zodat deze begrijpelijk en inzichtelijk zijn voor zowel de klant als het ontwikkelteam. Het verzamelen, analyseren en uitwerken van requirements wordt het analyse- en ontwerpproces genoemd. Waar de

Watervalmethode altijd erg gedetailleerd omging met informatieanalyse, functioneel ontwerp en technisch ontwerp, dicteert Scrum hier geen aanpak voor. Bij Scrum worden requirements doorgaans beschreven in de vorm van een User Story. Toch blijken bepaalde onderdelen essentieel om te komen tot een goede Product Backlog. Voorbeelden hiervan zijn bijvoorbeeld bepaalde modellen, zoals het opstellen van een Flowchart om het proces te beschrijven.

Voor User Stories wordt er meestal een Definition of Ready opgesteld. Dit is een omschrijving van eisen waaraan de User Story moet voldoen voordat deze kan worden opgenomen in een sprint. Bij grote en complexe projecten ziet Quintor echter dat User Stories niet volstaan. User Stories hebben als doel het snel duidelijk maken van een requirement in begrijpelijke taal. De User Stories vormen een gemeenschappelijke taal voor alle bij het project betrokkenen. Een ontwikkelaar heeft in sommige gevallen niet voldoende aan alleen een User Story maar moet een uitgebreide beschrijving hebben over wat het systeem moet kunnen en wat er moet worden gebouwd.

### **Probleem**

Bij het uitvoeren van softwareontwikkelprojecten is een goede Product Backlog essentieel. Om te komen tot een Product Backlog moet er een analyse- en ontwerpproces worden uitgevoerd. Omdat Scrum hiervoor geen aanpak dicteert wil Quintor hiervoor een gestandaardiseerde aanpak ontwikkelen, waarbij de methodiek voor het opstellen van een Product Backlog wordt beschreven en de manier waarop de daarin opgenomen User Stories 'ready' gemaakt kunnen worden.

Door het ontbreken van een methode op het gebied van analyse en ontwerp ziet Quintor bij klanten die overgaan van de traditionele manier van werken op Scrum dat er een gat ontstaat. Scrum geeft namelijk vooral richtlijnen over de werkwijze en de realisatie. Hierdoor ontstaan de volgende problemen:

- Er wordt te snel aan een project begonnen waardoor er later problemen ontstaan wegens het gebrek aan requirements en de uitwerking hiervan. Hierdoor komt er op dit gebied te veel verantwoordelijkheid te liggen bij de ontwikkelaar.
- Door het ontbreken van een invulling voor analyse en ontwerp wordt er te veel teruggegrepen naar de traditionele manier van werken waardoor een project te lang in het voortraject blijft hangen.
- Er ligt niet vast wat de rol van een informatieanalist in een Scrumproject is. Dit komt door de onduidelijkheid over de invulling van de analyse- ontwerpfase en het feit dat Scrum deze rol niet beschrijft. Bij grote en complexe projecten is een informatieanalist onmisbaar. Het achterhalen, specificeren, vastleggen en verduidelijken van de WAT-vraag is het domein van een informatieanalist. Het uiteindelijke doel van een analist is een begrippenkader te creëren over de WAT-vraag. De informatieanalist zorgt ervoor dat iedereen deze begrijpt, zowel de klant als de ontwikkelaars en testers.

## **Belangen**

Quintor is werkzaam bij de klant en daarom worden deze problemen door zowel Quintor als de klant ervaren. Beide partijen hebben belang bij een goed proces voor analyse en ontwerp binnen een Scrumproject, met duidelijke artefacten.

Belang Quintor: door een goed proces zal Quintor beter in staat zijn om hun klanten te ondersteunen wanneer deze beginnen met werken volgens de Scrummethode of al werkzaam zijn volgens Scrum. Quintor staat voor professionele softwareontwikkeling. Een adequaat analyse-en ontwerpproces is daar een onverbreekelijk onderdeel van. Dit proces zorgt er namelijk voor dat de ontwikkelde software de juiste IT-ondersteuning biedt.

Belang klant: de klant zal met behulp van een duidelijke invulling makkelijker over kunnen gaan van de traditionele manier van werken op Scrum of als ze al Scrum gebruiken in staat zijn om in hun huidige Scrumprojecten effectiever te gaan werken.

## **Analyse en ontwerp**

Analyse en ontwerp is een breed begrip en kan worden opgedeeld in een WAT- en een HOE-vraag. Analyse en ontwerp gaan in dit onderzoek over de WAT-vraag. De WAT-vraag geeft antwoord op wat de klant wil, welke functionaliteiten er nodig zijn en daarmee wat er gebouwd moet worden. De HOE-vraag gaat over hoe de functionaliteit technisch gerealiseerd gaat worden en is buiten de scope van dit onderzoek. Het analyse- en ontwerpproces betreft in deze opdracht alles dat moet gebeuren om de WAT-vraag van de klant zo goed mogelijk te achterhalen, specificeren, vastleggen en verduidelijken zodat alle stakeholders het begrijpen en er feedback op kunnen geven.

## **Oplossingsrichtingen**

In de praktijk worden bij Quintor en klanten van Quintor diverse aanpakken gebruikt om te komen tot een Product Backlog en om Product Backlog items(User Stories) ready te maken. Bijvoorbeeld door het opstellen van Use Cases, het uitvoeren van traditionele informatieanalyse, het maken van traditionele documentatie of een zelf bedacht document formaat.

## **Doel van het project**

Quintor is van mening dat om tot een juiste invulling voor analyse en ontwerp binnen Scrum te komen dit gebaseerd moet zijn op theorieën, ervaringen en praktijksituaties. Het doel van deze opdracht is het in kaart brengen van beschreven theorieën en de werkwijzen die organisaties hanteren met betrekking tot het komen tot de Product Backlog, het ready maken van User Stories en de rol van de informatieanalist hierin. Aan de hand hiervan kunnen conclusies en aanbevelingen worden geformuleerd. Quintor vindt het belangrijk dat er tijdens het onderzoek, naast aandacht voor knelpunten, ook aandacht voor succesfactoren is. Het achterliggende doel is dat de aanbevelingen en conclusies uit dit onderzoek input kunnen leveren om tot een gestandaardiseerde aanpak te komen.

### 1.3 Doelstelling

Dit onderzoek bevindt zich in de tweede fase van de interventiecyclus. De interventiecyclus bestaat uit een aantal fasen die elkaar opvolgen wanneer er sprake is van het oplossen van een probleem. Het model bestaat uit de volgende fasen: probleemanalyse, diagnose, ontwerp, interventie en evaluatie. In dit onderzoek is er sprake van een probleem dat erkend is bij de betrokken partijen. Het onderzoek bevindt zich om deze reden in de diagnostische fase. In deze fase volgt een onderzoek naar de oorzaken en achtergronden van het probleem. (Verschuren & Dorenwaard, 2007)

De doelstelling die geformuleerd is op basis van het projectkader luidt als volgt:

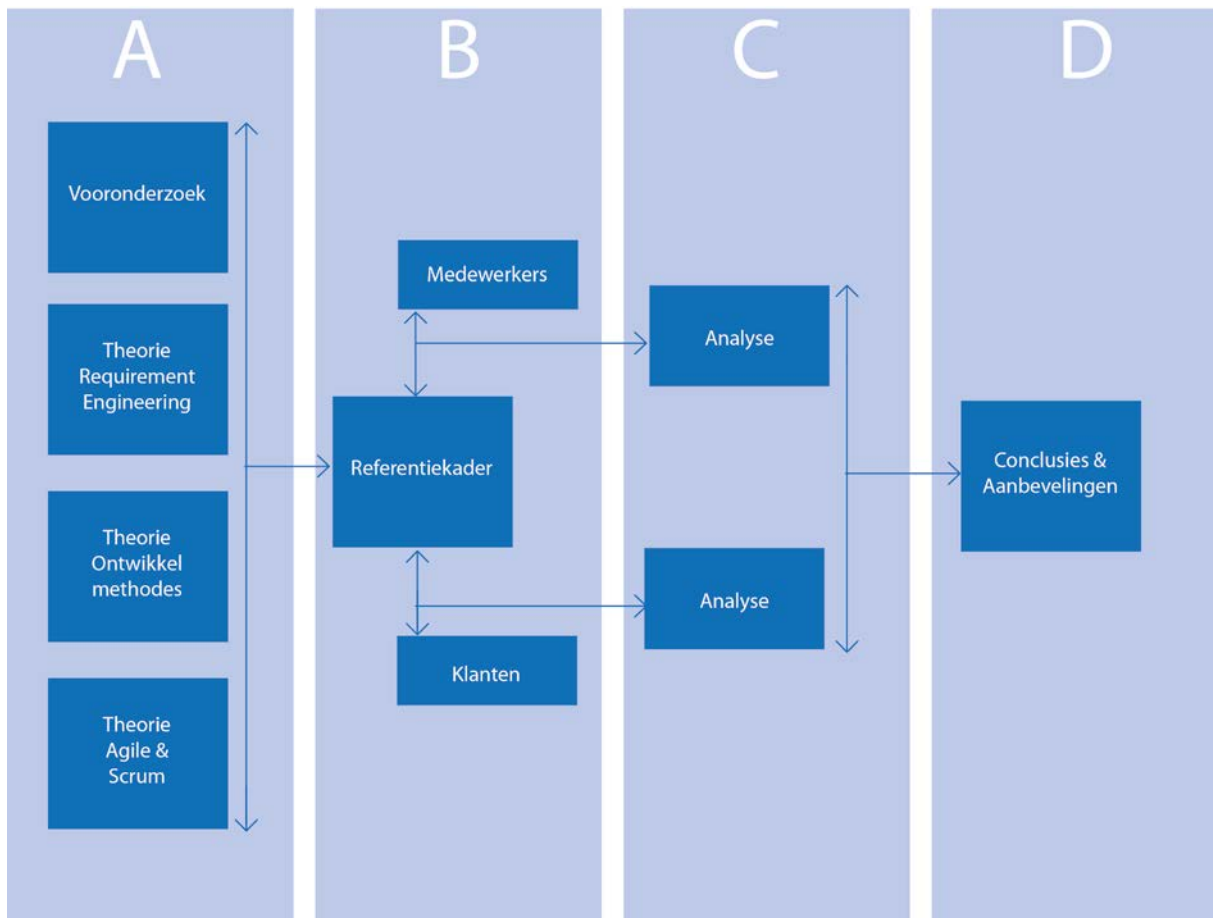
Het doel van het onderzoek is aanbevelingen doen aan Quintor ter verbetering van het analyse- en ontwerpproces in Scrumprojecten

*door*

op basis van een vooronderzoek en vastgestelde theorieën (over: requirement engineering, ontwikkelmethodes, Agile en Scrum) te komen tot een referentiekader om deze vervolgens te confronteren met klanten van Quintor en medewerkers van Quintor die werkzaam zijn bij deze klanten. Op deze manier zal inzicht worden gegeven in de werkwijze die zij hanteren met betrekking tot analyse en ontwerp in Scrumprojecten. Hierbij zal aandacht zijn voor zowel succesfactoren als knelpunten die optreden in praktijksituaties. Een beoordeling van de analyseresultaten die voortvloeien uit dit onderzoek zal bijdragen aan de oplossing van het probleem.



## 1.4 Onderzoeksmodel



### Verwoording

**(A)** Een bestudering van problemen die ontstaan in het voortraject van Scrum en richtingen waarin oplossingen worden gezocht, gebaseerd op gesprekken met medewerkers van Quintor (vooronderzoek) en een onderzoek naar theorieën (theorie requirement engineering, ontwikkelmethodes, Agile en Scrum), levert een referentiekader. **(B)** Door middel van een confrontatie tussen dit referentiekader en de onderzoeksobjecten (klanten van Quintor en medewerkers van Quintor die werkzaam zijn bij deze klanten) wordt de werkwijze in de praktijk in kaart gebracht en komen knelpunten en succesfactoren uit praktijksituaties aan het licht. **(C)** Een beoordeling van deze werkwijze in relatie tot het referentiekader resulteert in **(D)** conclusies en aanbevelingen ter verbetering van het analyse- en ontwerpproces in Scrumprojecten.

## 1.5 Centrale vragen

De centrale vragen zijn gebaseerd op het onderzoeksmodel. Het onderzoeksmodel bestaat uit vier fases (A, B, C en D). De drie centrale vragen zijn afgeleid uit het A, B, en C deel. Het antwoord op de drie centrale vragen helpt de doelstelling van het onderzoek te behalen. Bij iedere centrale vraag is een aantal deelvragen van toepassing. Het antwoord op de deelvragen helpt antwoord geven op de bijbehorende centrale vraag. (Verschuren & Dorenwaard )

### ***Centrale vraag 1: Hoe ziet de het referentiekader eruit gelet op het vooronderzoek en de geraadpleegde theorieën?***

#### *Vooronderzoek*

- a. Welke problemen spelen er rondom Scrum in relatie tot analyse en ontwerp?
- b. In welke richtingen worden er oplossingen gezocht?
- c. Welke actoren spelen een rol en welke belangen hebben zij?
- d. Waarom is er onderzoek nodig?

#### *Theoretisch kader*

- e. Hoe ziet (traditionele) requirement engineering er uit?
- f. Welke ontwikkelmethoden bestaan er en welke zijn relevant voor dit onderzoek?
- g. Wat wordt verstaan onder Agile?
- h. Wat wordt verstaan onder Scrum?

### ***Centrale vraag 2: Hoe wordt in de praktijk een invulling gegeven aan analyse en ontwerp binnen Scrum gelet op het beschreven referentiekader?***

#### *Empirisch onderzoek*

- a. Wat is de mening van klanten in relatie tot het beschreven referentiekader?
- b. Wat is de mening van medewerkers in relatie tot het beschreven referentiekader?

### ***Centrale vraag 3: Wat leert ons de beoordeling van de analyseresultaten met het oog op het doen van aanbevelingen op het gebied van analyse en ontwerp in een Scrumproject?***

#### *Analyse resultaten*

- a. Op welke manier kunnen deze inzichten bijdragen aan een mogelijke oplossing?
- b. Welke aanbevelingen kunnen er worden gedaan op basis van het onderzoek?

## 1.6 Methodologie & Strategie

Op basis van de centrale vragen is er een onderzoeksstrategie bepaald. De strategie betreft de vorm van het onderzoek. Aan de hand van de strategie is er voor een onderzoeksmethode gekozen. Vervolgens is voor de eerste twee centrale vragen de onderzoeksstrategie opgedeeld in verschillende fasen. De eerste fase bestaat uit een kwalitatief vooronderzoek en deskresearch. De tweede fase bestaat uit een kwalitatieve vergelijkende casestudy. Deze keuze is gemaakt om te waarborgen dat de verkregen resultaten ook theoretisch onderbouwd kunnen worden en vice versa.

### ***Centrale vraag 1: Hoe ziet het referentiekader eruit gelet op het vooronderzoek en de geraadpleegde theorieën?***

In de eerste fase, fase A in het onderzoeksmodel, is er sprake van twee onderzoeksstrategieën. Als eerste is er een vooronderzoek uitgevoerd. Hierin is door middel van informele ongestructureerde interviews met de opdrachtgever en een aantal werknemers van Quintor in kaart gebracht welke problemen er spelen en welke oplossingen er al zijn vanuit Quintor. Met de resultaten uit de interviews is het projectkader opgesteld. Tevens zijn de resultaten uit het vooronderzoek gebruikt om het referentiekader te vormen. De volgende mensen zijn geïnterviewd tijdens het vooronderzoek:

Johan Tillema (opdrachtgever)	CEO Quintor
Jan van Beek (begeleider)	COO Quintor
René Boere	CTO Quintor
Jan Willem Omlo	informatieanalist Quintor

In het tweede deel van de eerste fase is er onderzoek gedaan naar relevante theorieën. Dit onderzoek is gebeurd door middel van deskresearch. Hiervoor is gebruik gemaakt van literatuur- en internetbronnen. De keuze voor de onderzochte onderwerpen is gebaseerd op de vraag van de opdrachtgever en de geformuleerde doelstelling. De volgende onderwerpen zijn onderzocht:

- Requirement engineering
- Ontwikkelmethodes
- Agile
- Scrum

Binnen deze onderwerpen is er gezocht naar achtergrondinformatie en waar mogelijk detailinformatie over de drie hoofdthema's van deze thesis die zijn afgeleid uit het projectkader. Deze hoofdthema's zijn gedurende het hele onderzoek de rode draad geweest. Alle resultaten, conclusies en aanbevelingen zijn in deze thesis aan de hand van de drie hoofdthema's weergegeven en toegelicht. De hoofdthema's zijn:

- Het komen tot de Product Backlog
- Het ready maken van User Stories
- De rol van de informatieanalist

***Centrale vraag 2: Hoe wordt in de praktijk een invulling gegeven aan analyse en ontwerp binnen Scrum gelet op het beschreven referentiekader?***

In de tweede fase, fase B in het onderzoeksmodel, is een kwalitatieve vergelijkende casestudy uitgevoerd. De keuze voor een casestudy is gebaseerd op de omvang van de onderzoeksobjecten, de mogelijkheid tot een arbeidsintensieve benadering en de wens om de diepte in te gaan in plaats dan de breedte. Het was belangrijk bij dit onderzoek om een totaalbeeld te krijgen en niet uitsluitend een aantal variabelen. Tijdens de casestudy is het referentiekader geconfronteerd bij verschillende klanten van Quintor en medewerkers van Quintor die werkzaam zijn bij deze klanten. Deze confrontatie is gebeurd door middel van diepte-interviews.

Omdat er niet één maar verschillende cases zijn bestudeerd is hier sprake van een vergelijkende casestudy. Hierin is de hiërarchische methode gebruikt. Deze methode bestaat uit twee fasen. In de eerste fase is er onderzoek gedaan naar verschillende projecten bij organisaties. Hier is ieder project als aparte case beschouwd. Per project is in ieder geval een medewerker van Quintor die werkzaam is bij de organisatie of een medewerker van de organisatie zelf geïnterviewd. Dit is gebeurd door middel van semigestructureerde interviews met open vragen. Deze manier van interviewen is gekozen omdat het veel ruimte geeft voor input van de respondent. Vooraf zijn er een aantal topics opgesteld en is er nagedacht over mogelijke vragen per topic. Dit heeft een bepaalde structuur gegeven en heeft ervoor gezorgd dat in ieder interview dezelfde onderwerpen zijn behandeld. In bijlage 1 is de topiclijst te vinden en de mogelijke vragen per topic.

De cases zijn in eerste instantie onafhankelijk van elkaar onderzocht. De resultaten verkregen uit de acht cases zijn op dezelfde manier verwerkt. In de tweede fase zijn de resultaten die zijn verkregen uit de eerste fase vergeleken en beoordeeld. Hierin is gezocht naar overeenkomsten en verschillen tussen de resultaten verkregen uit de verschillende cases. Het uiteindelijk verkregen resultaat uit de casestudy is beoordeeld en vergeleken met het opgestelde referentiekader. Op basis van deze analyse zijn de uiteindelijke conclusies en aanbevelingen geformuleerd. (Verschuren & Dorenwaard, 2007)

### **Populatie**

In overleg met de opdrachtgever is er een keuze gemaakt voor de respondenten en de organisaties. Om generaliseerbare onderzoeksresultaten en een breed beeld te krijgen is ervoor gekozen om organisaties en projecten van verschillende omvang te benaderen. De helft van de projecten bestond uit meerdere teams en de andere helft bestond uit één team. Wat betreft de functie van de respondenten is er hoofdzakelijk geselecteerd op informatieanalist en/of functioneel ontwerper omdat deze het meest in relatie staan met het onderwerp dat deze thesis behandelt. Wanneer deze mensen niet beschikbaar waren is er gekozen voor een Product Owner. Onderstaande tabel toont een overzicht van de bedrijven, het aantal cases, de functie van de respondenten en het aantal respondenten per case.

Organisatie	Aantal behandelde projecten	Functie	Aantal respondenten
Ministerie van Economische Zaken	1	informatieanalist/functioneel ontwerper	1
CJIB	2	informatieanalist & informatieanalist/functioneel ontwerper	2
Fa-med	1	informatieanalist/functioneel ontwerper	2
Duo	1	informatieanalist & functioneel ontwerper	2
ING	1	ontwikkelaar/Product Owner	1
AEGON	1	informatieanalist/developer	1
Rabobank	1	informatieanalist	1

tabel 1 populatie

Ieder project is als afzonderlijke case beschouwd. Zo is er dus bij één organisatie sprake van twee behandelde cases. Om de respondenten te benaderen is er contact gelegd via de e-mail met medewerkers van Quintor en is hun gevraagd om een afspraak. Tevens is hun de vraag gesteld of zij ook nog iemand kenden binnen de organisatie die relevant was voor dit onderwerp. Op deze manier kon er snel contact worden gelegd en werden de juist mensen direct benaderd.

### Data-analyse

De diepte-interviews zijn opgenomen met een recorder. Vervolgens zijn alle interviews uitgewerkt in transcripten. Er zijn verschillende manieren om een transcriptie te maken. Voor dit onderzoek is de keuze gemaakt voor een globale transcriptie. Dit houdt in dat de opname is beluisterd en dat er is geprobeerd dit zo goed mogelijk uit te schrijven. Bij een globale transcriptie wordt er niet geprobeerd om alles zo volledig mogelijk en foutloos te noteren. Dit betekent dat niet relevante tekst zoveel mogelijk is weggelaten en dat gemaakte fouten tijdens het transcriberen niet worden verbeterd. Deze manier van transcriberen wordt veel gebruikt bij kwalitatief onderzoek omdat het snel een bruikbaar resultaat geeft zodat er voldoende tijd overblijft om de resultaten te analyseren. (Globale transcriptie, z.d.) Er is afgesproken met de respondenten om de opnames en transcripties vertrouwelijk te houden. Om deze reden is dit ook niet opgenomen als onderdeel van deze thesis maar bestaan deze alleen in de vorm van een vertrouwelijke externe bijlage. Deze vertrouwelijke externe bijlage kan alleen op aanvraag worden ingezien.

Omdat gegevens in de vorm van transcripten gecompliceerde gegevens zijn, is het voor dit onderzoek belangrijk geweest om een goede manier van data-analyse toe te passen. Tijdens de analyse was het daarom van belang om de gegevens in te delen en te condenseren. Gegevens kunnen benaderd worden vanuit een deductieve of inductieve benadering. Wanneer er sprake is van een deductieve benadering wordt ervoor gekozen om een bestaande theorie te gebruiken en aan de hand daarvan, vanuit de transcripten, codes en categorieën te destilleren. Bij een inductieve benadering wordt er getracht door middel van analyse een theorie op te bouwen aan de hand van de resultaten. In dit onderzoek is er een combinatie van beide benaderingen gemaakt door in eerste instantie de resultaten inductief te benaderen en vervolgens deductief te verwerken. Dit is gedaan door in eerste instantie categorieën te vormen en

deze af te leiden uit de resultaten in plaats van vanuit de theorie of het referentiekader. Hierdoor was er aandacht voor alle besproken onderwerpen. Deze manier van coderen heet open-coderen en houdt in dat delen tekst die van betekenis zijn voor het onderzoek voorzien worden van een code. Er is gekozen om zoveel mogelijk in-vivo codes te gebruiken. Dit zijn codes die zijn afgeleid vanuit woorden die de respondent heeft gebruikt. In de tweede stap zijn de codes in categorieën onderverdeeld. Wederom zijn hier de categorieën afgeleid vanuit de gegevens en niet vanuit het referentiekader en/of de theorie. Om overzicht te krijgen in de diverse categorieën zijn deze vervolgens ondergebracht in thema's. Deze thema's zijn wel afgeleid vanuit de opdracht en het referentiekader. Er is gekozen om pas in dit stadium de bevindingen te koppelen aan thema's om zo oog te hebben voor alle resultaten en bevindingen. Op deze manier is er gewaarborgd dat alle relevante bevindingen zijn meegenomen tijdens de analyse. Tijdens de analyse is steeds de tekst in de transcripten nagelezen. Dit kon makkelijk worden gedaan doordat de in-vivo code ook uit een nummer bestond dat verwees naar een bepaald transcript en een regelnummer in het transcript. Vervolgens is er per thema een tabel opgesteld met de belangrijkste bevindingen per case. Deze bevindingen zijn vervolgens behandeld en toegelicht in het dit onderzoeksrapport. (Saunders e.a., 2011)

### **Betrouwbaarheid en validiteit**

De gekozen methode voor gegevensverzameling en data-analyse heeft betrekking op de betrouwbaarheid en validiteit van de onderzoeksresultaten. Dit onderzoek betreft een kwalitatief onderzoek. Dit betekent dat de betrouwbaarheid en validiteit op een andere manier worden gewaarborgd dan wanneer het een kwantitatief onderzoek betreft. (Aken & Andriessen, 2011)

Betrouwbaarheid heeft betrekking op consistentie van de bevindingen. Een betrouwbaar onderzoek houdt in dat wanneer het onderzoek herhaald zou worden, een andere onderzoeker dezelfde onderzoekresultaten zou verkrijgen. Validiteit van het onderzoek houdt in dat de onderzoekresultaten ook daadwerkelijk gaan waarover ze lijken te gaan. Het kan namelijk zo zijn dat er factoren van invloed zijn die er voor zorgen dat er een misleidend effect ontstaat op de resultaten. (Saunders e.a., 2011) In dit onderzoek is er een aantal keuzes gemaakt om zo de betrouwbaarheid en validiteit zo goed mogelijk te waarborgen. Ten eerste is gedurende het praktijkonderzoek dezelfde manier van interviewen gebruikt en zijn de interviews door dezelfde persoon afgenomen. De interviews zijn allemaal in een afgesloten ruimte gehouden. De respondenten zijn geselecteerd op bepaalde relevantie. Het werken in een bepaald soort project en de rol die de respondent vertolkte in het project was hiervoor bepalend. Omdat er werd gevraagd kritisch en eerlijk te zijn was het belangrijk om te waarborgen dat de respondenten vrijuit spraken. Hierom is er afgesproken de namen en transcripten niet letterlijk op te nemen in deze thesis maar deze als vertrouwelijke bijlage te laten bestaan.

Doordat er voorafgaand aan de casestudy een theoretisch kader is opgesteld wordt de validiteit gewaarborgd. Overeenkomsten leiden op die manier tot extra validiteit van de gegevens. Voor de interviews zijn er vooraf topics en mogelijke vragen opgesteld die zijn afgeleid vanuit de literatuur en de vraag van de opdrachtgever. Deze topics zijn de leidraad van de gesprekken geweest om op deze manier zoveel mogelijk dezelfde gespreksonderwerpen te krijgen. Tijdens het praktijkonderzoek zijn meerdere cases behandeld. Om de validiteit te waarborgen zijn de resultaten uit alle cases op dezelfde manier

verwerkt. (Verschuren & Dorenwaard, 2007) Na een aantal interviews kwam in grote lijnen een aantal resultaten overeen. Dit ging met name over het hanteren van een bepaalde werkwijzen en een aantal overeenkomstige factoren die deze werkwijze beïnvloeden. Wanneer er sprake is van overeenkomsten vanuit verschillende interviews is er spraken van convergentie. Daarnaast was er ook sprake van een aantal verschillen. Dit is logisch aangezien verschillende metingen ook verschillende resultaten opleveren. Er is dan sprake van discriminerende validiteit. (Aken & Andriessen, 2011) Deze vormen van validiteit zijn bereikt door zoveel mogelijk verschillende bronnen te gebruiken.

De beschreven methode van data-analyse heeft ook betrekking op de betrouwbaarheid en validiteit. Om de betrouwbaarheid en validiteit ook tijdens de data-analyse te waarborgen is er gebruik gemaakt van het maken van een opname tijdens de diepte-interviews en zijn de opname uitgewerkt in transcripten. De transcripten zijn gecodeerd en gecategoriseerd. De transcripten zijn meerdere keren nagelezen en geraadpleegd om de verkregen resultaten vanuit de analyse te valideren. (Saunders e.a., 2011 )

## 1.7 Vooronderzoek

Het daadwerkelijke vooronderzoek is te vinden in de externe bijlage: *Vooronderzoek & Theoretisch kader*. Omdat Quintor al lang met het probleem dat deze thesis behandelt te maken heeft bestaat er bij Quintor al een aantal richtingen waarin oplossingen worden gezocht. Naast het vormen van het projectkader is het vooronderzoek daarom ook gebruikt om het referentiekader op te stellen zodat dit gebruikt kon worden tijdens de casestudy. Het resultaat uit het vooronderzoek dat is opgenomen in het referentiekader staat beschreven in hoofdstuk 2. Met het vooronderzoek zijn de volgende deelvragen beantwoord:

Deelvragen vooronderzoek-eerste centrale vraag

- a. Welke problemen spelen er rondom Scrum in relatie tot analyse en ontwerp?
- b. In welke richtingen worden er oplossingen gezocht?
- c. Welke actoren spelen een rol en welke belangen hebben zij?
- d. Waarom is er onderzoek nodig?



## 1.8 Theoretisch kader

In het theoretisch kader is een aantal theorieën behandeld die uitgebreid staan beschreven in de externe bijlage: *Vooronderzoek & Theoretisch kader*. Het belangrijkste resultaat van het theoretisch kader is te vinden in hoofdstuk 2. Het theoretisch kader is samen met het vooronderzoek gebruikt om het referentiekader op te stellen. In deze paragraaf staat beschreven welke theorieën er zijn onderzocht en waarom deze relevant zijn voor het onderzoek.

**Requirement engineering:** De theorie over requirement engineering betreft het totale proces van het verzamelen en vastleggen van requirements. Dit betreft hoofdzakelijk de traditionele werkwijzen. Deze achtergrond is relevant voor deze thesis omdat bepaalde delen van dit proces ook binnen Scrum toepasbaar zijn en het de taken van een informatieanalist beschrijft.

**Ontwikkelmethoden:** Scrum is een van de vele ontwikkelmethoden die er bestaan. Inzicht in verschillende methoden en vooral de manier waarop verschillende methoden omgaan met analyse en ontwerp is relevant voor deze thesis.

**Agile:** Agile is de methodologie waar het Scrumproces op is gebaseerd. De principes die Agile voorschrijft vormen het uitgangspunt tijdens softwareontwikkeling volgens Scrum. Het begrijpen en vastleggen van deze methodologie is daarom belangrijk voor de thesis.

**Scrum:** Voor deze thesis is het van belang dat er een duidelijk beeld bestaat over wat Scrum inhoudt en hoe het proces wordt uitgevoerd. Vanuit de theorie wordt het Scrumproces beschreven en is er gezocht naar beschreven aanvullingen op de Scrummethode.

Met het theoretisch kader zijn de volgende deelvragen beantwoord:

Deelvragen theoretisch kader-eerste centrale vraag

- e. Hoe ziet (traditionele) requirement engineering er uit?
- f. Welke ontwikkelmethoden bestaan er en welke zijn relevant voor dit onderzoek?
- g. Wat wordt verstaan onder Agile?
- h. Wat wordt verstaan onder Scrum?

# Hoofdstuk 2

## Resultaten vooronderzoek en theoretisch kader

*Door het theoretisch kader en het vooronderzoek samen te voegen is het referentiekader gevormd. In paragraaf 2.1 wordt per hoofdthema aangegeven welke delen uit het theoretisch kader en het vooronderzoek worden opgenomen in het referentiekader. In paragraaf 2.2 worden de punten uit het referentiekader die zijn afgeleid uit paragraaf 2.1 per hoofdthema weergegeven. Op deze manier kan worden aangetoond hoe het referentiekader tot stand is gekomen en op welke theorie en/of bevindingen het is gebaseerd. Dit hoofdstuk geeft antwoord op de eerste centrale vraag:*

*Centrale vraag 1: Hoe ziet het referentiekader eruit gelet op het vooronderzoek en de geraadpleegde theorieën?*

## 2.1 Weergave resultaten vooronderzoek & theoretisch kader

### 2.1.1 Komen tot de Product Backlog

Uit de theorie blijkt dat het proces om alle requirements te verkrijgen en te specificeren, traditioneel wordt aangeduid met de term Requirement Engineering. Dit proces bestaat uit de volgende fases: eliciteren, analyse, specificatie en validatie. (Sommerville, 2011) Er is gebleken dat er weinig theorie bestaat over hoe een soortgelijk proces kan worden doorlopen om tot de Product Backlog te komen binnen Scrum. Wat wel blijkt is dat dit het best kan worden gedaan wanneer het Scrumteam samen met de stakeholders op basis van een idee of een visie brainstormt om tot de juiste items te komen. Hierbij is het belangrijk om de focus te leggen op de meest noodzakelijke functionaliteit en deze te formuleren als User Stories. Op deze manier komt de (initiële) Product Backlog tot stand. (Pichler, 2010a) Daarnaast zijn er nog verschillende manieren gebleken om tijdens het project de Product Backlog te onderhouden. Dit gebeurt meestal in de vorm van Backlog Refinement<sup>1</sup>. (Schwaber & Sutherland, 2011)

Uit het vooronderzoek blijkt dat Quintor zelf een aanpak hanteert om te komen tot de Product Backlog in de vorm van een aantal stappen die worden doorlopen samen met stakeholders:

1. Doel van het project bepalen(visie)
2. Actoren bepalen die met het systeem interacteren
3. Functies per actor in kaart brengen
4. Functies prioriteren
5. Belangrijke processen beschrijven zodat tijdens het vormen van User Stories er functionaliteiten geformuleerd kunnen worden
6. Interfaces bepalen zodat duidelijk is met welke systemen het project te maken heeft
7. User Stories formuleren
8. Opstellen Product Backlog, prioriteren en inschatten

Deze stappen geven in feite een concrete invulling aan de beschreven theorie waarin met stakeholders op basis van een visie, belangrijke functionaliteiten worden benoemd en deze vertaald worden naar User Stories. Tijdens het proces visualiseert Quintor belangrijke en complexe processen. Deze legt Quintor voor aan de stakeholders. Het liefst in sessies waarin de resultaten met elkaar besproken worden en van commentaar worden voorzien. Op basis van de theorie en het vooronderzoek wordt er aangenomen dat bovenbeschreven stappen noodzakelijk zijn om te komen tot een Product Backlog.

#### **Voortraject**

Met alleen het opstellen van de Product Backlog is er nog niet voldoende gedaan om ook te kunnen beginnen met de eerste sprint. Er blijkt hiervoor vaak een zogenaamde sprint 0 te worden ingericht. Kort gezegd doe je in sprint 0 alles wat je moet doen om goed voorbereid aan een Scrumproject te kunnen beginnen. In hoofdstuk 5 van externe bijlage: vooronderzoek & theoretisch kader, is een aantal mogelijke onderdelen genoemd dat in sprint 0 kan voorkomen. Gezien de Scope van dit onderzoek worden niet al

<sup>1</sup> Wordt ook wel Backlog Grooming genoemd.

deze onderdelen opgenomen in het referentiekader. Naast het doorlopen van de bovenstaande stappen om te komen tot de Product Backlog zijn verder in een sprint 0, het opstellen van de Definition of Ready en Definition of Done en het ready maken van User Stories voor 1 à 2 sprints van belang. (Anders, 2013 ; Architech Solutions, 2010)

### 2.1.2 Ready maken

#### Flow naar ready

Naast het ready maken van items als onderdeel van sprint 0 blijkt het ready maken van items een continu proces tijdens een sprint. Dit kan worden beschouwd als een parallelle sprint waarin items ready gemaakt worden voor de volgende sprint. Dit wordt de Flow naar ready genoemd. Het is de verantwoordelijkheid van de Product Owner om items ready te krijgen en omdat in grote organisaties en / of projecten dit veel werk kan zijn, dient hij hierin ondersteund te worden door informatieanalisten. De Flow naar ready kan inzichtelijk gemaakt worden op een board (zoals bij een scrumboard). Het doel hiervan is dat iedereen zicht heeft op de items waaraan wordt gewerkt en welke er dus aan zitten te komen voor de volgende sprint. De categorie-indeling voor dit board kan als volgt zijn: New, Preparing, Ready en in Sprint. (Beaumont, 2009)

#### Validatie

Aangezien er uit de theorie is gebleken dat het lastig is om te bepalen wat voldoende informatie is voor een ontwikkelteam om aan de slag te kunnen, (Paetsch e.a.,2003) wordt er aangenomen dat er tijdens of na de Flow naar ready moet worden gecheckt of een ready User Story voldoende input is voor het ontwikkelteam om aan de slag te kunnen tijdens een sprint. In het traditionele proces van Requirement Engineering was validatie namelijk ook een belangrijk onderdeel. Wanneer toch blijkt dat dit niet voldoende is tijdens een sprint moet daar binnen de Flow naar ready op worden ingespeeld zodat dit niet uitsluitend de verantwoordelijk van het ontwikkelteam wordt.

#### Juiste balans

Zowel overspecificatie als onderspecificatie wordt vanuit de theorie als waste beschouwd. (Leffingwell, 2011 ; Poppendieck & Poppendieck, 2003) Het blijkt lastig om de hoeveelheid waste te beperken maar tegelijkertijd wel genoeg te doen zodat het ontwikkelteam voldoende input heeft tijdens een sprint. (Paetsch e.a.,2003) Dit is ook te zien aan de problemen zoals deze zijn benoemd tijdens het vooronderzoek en staan beschreven in het projectkader. Het resultaat is vaak dat er of teveel of te weinig op het gebied van analyse en ontwerp gebeurt. Het is dus belangrijk om een balans tussen deze twee uitersten te vinden. Om de diepte van analyse en ontwerp te bepalen is het belangrijk om traditionele documenten en procedures los te laten en in plaats daarvan de focus te leggen op hoe het systeem zich dient te gaan gedragen. (Cottmeyer & Henson z.d.) Het zoeken naar een balans wordt binnen Agile/Scrum met de termen 'Just-in-Time' en 'Just-Enough' aangeduid. User Stories moeten 'Just-Enough worden gespecificeerd en 'Just-in-time' klaar zijn aan het begin van een sprint. (Sutherland & Schwaber 2007)

#### Overspecificatie

Het blijkt dat een invulling voor analyse en ontwerp vaak wordt gezocht in traditionele methoden zoals de Watervalmethode. Er is duidelijk geworden dat de Watervalmethode en Agile veel tegenstrijdigheden

bevatten. (Agile introduction, 2008) Op basis hiervan wordt aangenomen dat het lastig is om een invulling voor analyse en ontwerp te zoeken in traditionele aanpakken. Het lijkt daarom logischer om dit te doen door middel van een meer Agile-achtige aanpak. Dit kan bijvoorbeeld door het inrichten van de eerder besproken sprint 0 en de Flow naar ready (Beaumont, 2009) om de hoeveelheid waste te beperken. Op deze manier kan worden voorkomen dat er onnodige processen worden doorlopen en documenten worden opgeleverd.

### **Onderspecificatie**

De theorie wijst uit dat het nodig is om User Stories begrijpelijk te formuleren zodat de klant deze goed begrijpt. Om het optimale uit het Scrumproces te halen moet dit ook op deze wijze gebeuren. De User Stories creëren een gemeenschappelijke taal die erg belangrijk is binnen Scrum. (Sutherland & Schwaber, 2007; Pichler, 2010a) Een veelgemaakte fout is dat de User Stories documentatie vervangen. (Arendsen e.a., 2008) Uit het vooronderzoek blijkt dat in kleine projecten het kan voorkomen dat een ervaren ontwikkelteam met alleen wat User Stories ook daadwerkelijk uit de voeten kan maar dat bij middelgrote en grote projecten dit vaak niet het geval is. Er kan worden geconcludeerd dat dan een te groot deel van de WAT-vraag in een sprint terecht komt en dus ook bij de ontwikkelaar. Het ontwikkelteam moet over voldoende specificatie beschikken om tijdens een sprint aan de slag te kunnen. Het is alleen lastig om te bepalen wat dat is. (Paetsch e.a., 2003) Dit betekent dat alleen tekst onvoldoende kan zijn en dat dit waar nodig, aangevuld kan worden met bijvoorbeeld een wireframe, diagrammen en het weergeven van workflows. (Pichler, 2012; Pichler, 2010a)

### **De Definition of Ready**

Er is gebleken dat de Definition of Ready een belangrijke aanvulling is op Scrum om ervoor te zorgen dat een User Story volledig en uitgebreid genoeg is. User stories dienen aan deze voorwaarden te voldoen voordat ze worden opgenomen in een Sprint. (Pichler, 2010b)

### **2.1.3 De rol van de informatieanalist**

Het is duidelijk geworden dat de rol van Product Owner erg belangrijk is binnen Scrum. De Product Owner is de persoon die de requirements verzamelt en prioriteert op basis van business value. Hij is daarin de persoon die verantwoordelijk is voor het beheren van de Product Backlog. De klant en eindgebruikers worden door deze Product Owner vertegenwoordigd. (Sutherland & Schwaber 2007) Zoals Scrum de rol beschrijft, vertoont deze veel overlap met de rol van een informatieanalist. Hoewel deze rol binnen Scrum niet wordt beschreven kan deze niet worden gemist. Het proces van elicitering, analyse, specificatie, validatie en management is volgens Peatch e.a. (2003) in bepaalde mate ook aanwezig tijdens het werken volgens de Scrummethode. Dit proces zou daarom een informatieanalist moeten doorlopen en niet een Product Owner. In de theorie worden drie mogelijke rollen voor de informatieanalist besproken namelijk: als Product Owner, naast de Product Owner of in het team. (Arendsen e.a., 2008) De plek en de rol van de informatieanalist kan dus verschillen per project. (Cottmeyer & Henson z.d.) Uit het vooronderzoek blijkt dat bij Quintor de informatieanalist vaak ook de functioneel ontwerper is bij Scrumprojecten. Uit het vooronderzoek blijkt dat dit vaak voorkomt bij Scrumprojecten omdat er niet meer gewerkt wordt in gescheiden fases. Het gaat veel meer om het doel dan de rol. De rol van de informatieanalist draait binnen Agile/Scrum veel meer om het faciliteren van mensen in tegenstelling tot de traditionele aanpak. De

informatieanalist moet de persoon zijn die de discussie tussen het team en de Product Owner mogelijk maakt. De informatieanalist is hiervoor de juiste persoon omdat deze beschikt over bepaalde systeemkennis en in staat is om requirements boven water te krijgen en te specificeren. (Cottmeyer & Henson z.d.) Op basis van het vooronderzoek en de theorie wordt geconcludeerd dat de rol van de informatieanalist en een functioneel ontwerper veel overlap vertonen omdat zij beide dienen om de WAT-vraag te beantwoorden. Deze rollen dienen veel meer geïntegreerd te worden uitgevoerd. Hierdoor is het denkbaar dat deze rollen door dezelfde persoon worden uitgevoerd. Zoals eerder is genoemd moeten requirements 'Just-Enough' worden gespecificeerd en 'Just-in-time' klaar zijn. (Sutherland & Schwaber 2007) Het is voor een informatieanalist daarom erg belangrijk om dit te kunnen doen. Daarnaast is de informatieanalist ook van belang tijdens het eerder genoemde Backlog Refinement. Met Backlog Refinement wordt bedoeld het opsplitsen van grotere User Stories in kleinere, het prioriteren van de Product Backlog en het toevoegen van nieuwe User Stories aan de Product Backlog. Er kan worden gesteld dat Backlog Refinement in bepaalde mate een invulling is van het traditionele requirements management. Daarom moet een informatieanalist hier samen met de Product Owner verantwoordelijk voor zijn. (Arendsen e.a., 2008)

## 2.2 Het referentiekader

De bevindingen uit paragraaf 2.1 zijn vertaald naar een referentiekader. Dit referentiekader is gebruikt tijdens de casestudy doordat hier vragen uit zijn afgeleid.

### 2.2.1 Komen tot de Product Backlog

#### Opstellen

- Om te komen tot de Product Backlog dienen de volgende stappen te worden doorlopen voorafgaand aan de eerste sprint:
  1. Visie bepalen
  2. Actoren bepalen
  3. Functies per Actor in kaart brengen
  4. Functies prioriteren
  5. Belangrijke processen beschrijven zodat tijdens het vormen van User Stories er functionaliteiten geformuleerd kunnen worden
  6. Interfaces bepalen zodat duidelijk is met welke systemen het project te maken heeft
  7. User Stories formuleren
  8. Opstellen Product Backlog, prioriteren en inschatten

#### Voortraject

- Er moet een sprint 0/voortraject plaats vinden waarin naast het opstellen van de Product Backlog de Definition of Ready en Definition of Done wordt bepaald en User Stories voor de eerste 1 à 2 sprints ready gemaakt worden.

### 2.2.2 Ready maken

#### Flow naar ready

- De User Stories dienen in verschillende categorieën worden ingedeeld om de fase waarin de User Story verkeert inzichtelijk te maken: New, Preparing, Ready en in Sprint.
- Het ready maken van User Stories dient als een apart proces te worden beschouwd en dient te worden gezien als een Flow naar ready. Op deze manier kan continu werkvoorraad worden gecreëerd voor de komende sprints.
- Dit proces dient te worden uitgevoerd voor de eerste sprint en tijdens sprints. Tijdens de sprint is dit een parallel proces aan de huidige sprint met als doel iedere keer weer voldoende stories ready te hebben als input voor de komende sprint.
- Het is ook tijdens sprints belangrijk om 1 à 2 sprints vooruit te werken en uitsluitend grote lijnen te bepalen en niet alles tot in detail uit te werken.
- Wanneer items toch onvoldoende ready blijken dient hierop geacteerd te worden, hetgeen kan leiden tot veranderingen in het proces waarin de Flow naar ready wordt uitgevoerd.

### **Validatie**

- Een User Story dient voldoende informatie te bevatten zodat een ontwikkelaar aan de slag kan. Om te bepalen wat dit is moet dit gecheckt worden.

### **Definition of Ready**

- De Definition of Ready moet worden opgesteld bij de start van het project en deze moet worden nageleefd zodat hier een algemeen begrip over heerst en alleen ready User Stories de sprint in gaan. Ready betekent User Stories conform deze definitie opleveren.

### **Specificatie**

- Om User Stories van voldoende informatie te voorzien kan gebruik worden gemaakt van hulpmiddelen zoals het opstellen van Wireframes en Flowcharts.
- User Stories moeten 'Just-Enough' worden gespecificeerd en 'Just-in-Time' klaar zijn aan het begin van een sprint.
- Een invulling van de specificatie moet niet gezocht worden in traditionele manieren van werken (de Watervalmethode) maar kan beter op een Agile/Scrum manier worden ingevuld om zo de hoeveelheid waste te beperken.
- Binnen Scrum moet, wanneer er een invulling wordt gezocht voor analyse en ontwerp, de focus niet liggen op het opleveren van documenten, maar op het hoe het systeem zich uiteindelijk moet gedragen.

### **2.2.3 De rol van de informatieanalist**

- De rol van een informatieanalist mag binnen Scrum niet worden vergeten. De Product Owner dient ondersteund te worden door een informatieanalist wanneer het project hier om vraagt.
- Het ready maken van items is de verantwoordelijkheid van de informatieanalist en de Product Owner.
- De rol van de informatieanalist en de functioneel ontwerper liggen bij Scrum dicht bij elkaar. Het is denkbaar dat deze rollen door dezelfde persoon worden vertolkt en ingevuld.
- De informatieanalist moet de persoon zijn die de discussie tussen het team en de Product Owner faciliteert. De informatieanalist is hiervoor de juiste persoon omdat deze beschikt over systeemkennis en in staat is om requirements boven water te krijgen en te specificeren.
- De informatieanalist is op verschillende plaatsen denkbaar namelijk: als onderdeel van het team, als Product Owner of als ondersteuning van de Product Owner.
- De informatieanalist is ook tijdens het project betrokken bij het managen van de Product Backlog in de vorm van Backlog Refinement.



# Hoofdstuk 3

## Resultaten casestudy

*De belangrijkste resultaten zijn in dit hoofdstuk per hoofdthema samengevat weergegeven. In bijlage 2 is een overzicht te vinden met een aantal belangrijke resultaten per case. Er is in dit hoofdstuk een aantal citaten toegevoegd om de resultaten te verduidelijken. Met de resultaten verkregen uit de casestudy wordt de volgende centrale vraag en bijbehorende deelvragen beantwoord:*

*Centrale vraag 2: Hoe wordt in de praktijk een invulling gegeven aan analyse en ontwerp binnen Scrum gelet op het beschreven referentiekader?*

- a. Wat is de mening van klanten in relatie tot het beschreven referentiekader?*
- b. Wat is de mening van medewerkers in relatie tot het beschreven referentiekader?*

### 3.1 Weergave resultaten casestudy

Tijdens de vergelijkende casestudy zijn veel onderdelen van het Scrumproces besproken en bleken veel van deze onderdelen een relatie met analyse en ontwerp te hebben. Zoals in hoofdstuk 1 staat beschreven zijn de transcripten gecodeerd en zijn deze vervolgens verdeeld in categorieën. Het resultaat hiervan is inzichtelijk gemaakt in bijlage 4. Er is toen gekeken welke categorieën onder de hoofdthema's vielen, welke categorieën samengevoegd konden worden en welke nieuwe er konden worden gedefinieerd. De uiteindelijke hoofdthema's en subthema's die zijn voortgekomen vanuit de analyse zijn weergegeven in bijlage 5.

#### 3.1.1 Komen tot de Product Backlog

Met betrekking tot het opstellen van de Product Backlog kan een duidelijk onderscheid gemaakt worden tussen het in eerste instantie vormen van de Product Backlog op Epic-niveau en het vormen van de Product Backlog gelijk op User Story-niveau. Wanneer er wordt gekeken naar de overige kenmerken die deze projecten hadden, blijkt dat wanneer er werd gekozen om op Epic-niveau te werken, er sprake was van grote projecten, met meerdere teams en duidelijke hiërarchische lagen in de organisatie. Hier was dan sprake van de Watervalmethode in het voortraject en Scrum in de realisatie. In twee gevallen betekende dit dat er ook een duidelijke scheiding was tussen informatieanalisten en functioneel ontwerpers.

#### Verschillende niveaus

In één case werden de Epic's beschreven op portfolio-niveau. Een niveau lager werden er van deze Epic's User Stories gemaakt. Op deze manier kwam de Product Backlog op verschillende niveaus tot stand. Een groot verschil was hier dat deze teams allemaal bij elkaar zaten en dat het beschrijven van Epic's gebeurde door een team dat betrokken was bij de realisatie en zelf ook in sprints werkte. Op deze manier konden Epic's op hoog niveau worden geformuleerd zonder een te grote scheiding te creëren tussen traditionele informatieanalyse en de realisatie.

#### Opstellen

Het komen tot de Product Backlog had in zes cases de overeenkomst dat het proces waarvoor de IT-ondersteuning moest komen high level werd doorlopen met stakeholders. Tijdens het doorlopen van het proces werden Epic's of User Stories benoemd. Met name de informatieanalist/functioneel ontwerper speelde hierin een grote rol. Bij de projecten waar dit op Epic-niveau gebeurde ging hier meestal een traditioneel proces aan vooraf. In twee cases ging het om een

*“Dus we zijn door die website heengegaan en hebben alle onderdelen op die website benoemd. Daar hebben we User Stories van gemaakt. Die hebben we op de Product Backlog gezet.”*  
Case 1

*“De meest recente ontwikkeling is dat we even het proces gaan doorlopen met mensen die er veel van weten en dat we op basis daarvan bepalen wat er allemaal aangepast moet worden. Dan heb je eigenlijk al een lijst met User Stories die je als basis kan gebruiken om verder te gaan.”*  
Case 6

verbeterslag op een bestaand product. In deze cases kwam naar voren dat eerder opgestelde Use Cases werden gebruikt om de processen te doorlopen en benoemen. Op deze manier werd de Use Case gebruikt om User Stories te formuleren.

### **Visueel artefact**

Bij drie projecten was er een high level visueel artefact/blauwdruk van het systeem opgesteld. De vorm varieerde van een klikbaar prototype, een procesplaat met Flowcharts tot een erg globale tekening van bepaalde processen. Opvallend was dat in een andere case, waar dit niet bestond, er wel een duidelijke wens was voor een soortgelijk iets. Er werd aangegeven dat een soortgelijk hulpmiddel erg goed helpt om het proces te doorlopen en aan de hand daarvan User Stories te benoemen. Tevens wordt het uiteindelijke doel inzichtelijk en kan men gedurende het project zien waar men zich begeeft in het proces. Op deze manier kan de voortgang getoond worden en kan men feedback leveren. Het High Level zijn van een dergelijk hulpmiddel werd benadrukt. Dit om te voorkomen dat het uiteindelijke doel te veel in detail wordt vastgelegd. Daarnaast was het hebben van een duidelijke visie een eis om dit visueel artefact (goed) te kunnen opstellen.

### **Voortraject**

In twee gevallen was er sprake van een als zodanig genoemde sprint 0. Hierin was er vooral sprake van het neerzetten van high level functionaliteiten. De term sprint 0 werd niet veel gebruikt om het voortraject mee aan te duiden. In drie gevallen is Backlog Refinement ter sprake gekomen. Hierin was de informatieanalist en de Product Owner betrokken.

*“Dat je die kapstok hebt die heel visueel is. Geen lappen tekst maar een visie op een half A4 met wat je wilt en daarnaast gewoon een visueel artefact. Dat is heel krachtig. Vanuit daar kan je dan heel goed je Product Backlog maken en bijhouden waar je bent.”*  
Case 1

*“Dat roep ik altijd op het moment dat je met Scrum of iteratieontwikkelen aan de slag gaat. Dan vind ik dat je uiteindelijk wel je doel ergens moet vastleggen. Dus je moet eerst globaal een blauwdruk maken waarin je aangeeft van we zitten nu hier en we gaan die kant op.”*  
Case 8

### 3.1.2 Ready maken

Het ready maken van User Stories voor een sprint is een proces gebleken dat erg varieert. Wat er tijdens dit proces gebeurde, hing sterk af van wat er onder ready werd verstaan. Wat de sprint in ging varieerde van een User story tot een compleet functioneel ontwerp in de vorm van een Use Case. De vorm hing vaak af van hoe de rollen waren verdeeld en hoe de rest van de organisatie was ingericht.

Wanneer een informatieanalist/functioneel ontwerper onderdeel was van een team dan liep deze vaak iets vooruit om wat werkvoorraad te creëren. De overige specificatie werd dan gedaan in de sprint. Ging het om een losstaand ready-team<sup>2</sup> dan werden User Stories ready gemaakt voor de sprint en ging het ready-team vervolgens verder met de voorbereidingen voor de komende sprint. Het ready-team bleef dan wel beschikbaar voor eventuele vragen maar maakte verder geen onderdeel uit van het ontwikkelteam. Hierdoor bleek er wel minder ruimte te zijn om in te kunnen spelen op extra behoefte vanuit de sprint.

Uiteindelijk draaide het er bij ready maken om dat het team aan de slag kon tijdens een sprint. Wat dit was werd vaak bepaald door een organisatiestandaard die voorschreef wat er gedocumenteerd moest worden of er werd per User Story met het team en andere stakeholders afgestemd wat zij nodig hadden. Soms was een documentatie-eis zo uitgebreid dat er bij de start van het project werd geprobeerd om dit zoveel mogelijk terug te brengen zodat er meer ruimte kwam voor Just-in-Time en Just-Enough analyse en ontwerp.

#### Specificatie

In iedere case speelde de Use Case een grote rol. Soms was de Use Case opstellen onderdeel van het ready maken en soms was dit onderdeel van de sprint. Het formaat van de Use Case liep uiteen van een volledig Functioneel Ontwerp tot slechts enkele onderdelen van een Use Case zoals het beschrijven van een scenario of het opstellen van een diagram. Ook dit kwam meestal door bepaalde documentatiestandaarden, een bepaalde cultuur of de manier waarop het ontwikkelteam de specificatie wilde hebben.

*“Het is gewoon heel afhankelijk van het team. Bijvoorbeeld twee teams, die wilde volledige documentatie hebben. De bouwering bij wijze van spreken niet aan de slag als de documentatie er niet was. Dat is vrij extreem” Case 2*

*“Afhankelijk van de complexiteit van de User Story was soms de User Story zelf al voldoende. Soms had je wat complexere processen waar je echt wel een Use Case-achtige opbouw nodig had. Dus de Use Case beschreef ik dan meestal bij de User Story als los attachment. Daarnaast maakte ik mock-ups van de schermen” Case 1*

*“Nou in Scrum is het niveau waarop je de analyse en ontwerp moet doen heel erg afhankelijk van de rest van het team dat je er hebt zitten. Heb je daar een team zitten met hardcore techneuten in een hele functionele omgeving dan moet je het veel verder uitwerken. Ergens heb je in elk team een paar toppertjes nodig die dat project op sleeptouw nemen. Dat kunnen ontwerpers zijn die heel ver gaan richting de techniek, dat kunnen techneuten zijn die heel ver gaan richting de functionele kant.” Case 5*

<sup>2</sup>Er wordt hier gekozen voor de term ready-team. Hiermee worden de mensen bedoeld die zich bezig houden met het ready maken van User Stories. Vaak wordt dit ook wel een kernteam genoemd en bestaat dit team niet uitsluitend uit informatieanalisten maar ook uit softwarearchitecten.

Het creëren van voldoende werkvoorraad werd in alle cases als lastig ervaren. In sommige gevallen uitte dit zich er in dat het realisatieteam op stop gezet moest worden of dat het realisatieteam zelf mee ging helpen met het ready maken van User Stories. Dit kwam vaak doordat men afhankelijk was van externe factoren zoals kennis in de organisatie van systemen, werken vanaf gescheiden locaties of van andere procedures die niet goed aansloten bij een iteratief ontwikkeltraject. Er bleken dan vaak veel onduidelijkheden en hierdoor werd het ready maken complexer. Een andere oorzaak was dat het ready-team onvoldoende tijd had vanwege betrokkenheid in andere projecten of in de lijn.

### **Validatie**

In alle gevallen werd er gevalideerd, of dat wat er beschreven stond, voldoende en correct was. Dit gebeurde zowel richting het ontwikkelteam als richting overige stakeholders. Hoe dit gebeurde liep uiteen van het bespreken van een User Story tijdens een meeting, tot complete reviewprocedures richting meerdere stakeholders. De informatieanalist en/of functioneel ontwerper was in deze gevallen degene die om de review vroeg. Ondanks dat het checken of iets ready was vrijwel in alle gevallen werd gedaan bleek het lastig om ook daadwerkelijk ready items de sprint in te krijgen. Wanneer de validatie iets minder formeel gebeurde werd er voor gekozen om dit onderdeel te maken van de Sprint Planning Meeting of de Backlog Refinement-sessie.

### **De Definition of Ready**

De Definition of Ready hield altijd in dat het voor iedereen duidelijk moest zijn wat er moest gebeuren. De bovenbeschreven validatie was vaak onderdeel van de Definition of Ready. In vrijwel iedere case werd aangegeven dat de hantering van de Definition of Ready beter kon door deze strakker te handhaven en beter te beschrijven. Er werd in vier gevallen aangegeven dat handhaving zou betekenen dat er geen werk in de sprint zou komen en dit werd als problematisch ervaren. Problemen rondom het handhaven van de Definition of Ready kwamen voort uit het feit dat deze niet realistisch was. De Defenition of Ready wordt bij de start van het project opgesteld. Men is in dit stadium geneigd om deze dan te beschrijven met te hoge eisen die in de praktijk niet haalbaar zijn.

*“De ontwerpers liepen redelijk gelijk met bouw en test. Natuurlijk wel iets voor want ze hadden vaak wel wat voorwerk gedaan maar dat was ook de frustratie soms, dat de ontwerpers de bouwers en testers niet echt konden bijhouden” Case 3*

*“Er moet wat voorraad zijn voor het done-team. Op dit moment merken we dat we ook die 2 weken werkvoorraad niet halen en dus zitten we in een traject waarbij we drie weken pre-ready werk doen en het done-team wordt even op pauze gezet. Op deze manier kunnen we wat voorraad creëren.” Case 8*

*“Er moet gewoon een enorme Product Backlog komen voor die teams. Nu is het vaak zo dat wanneer er iets bij komt dit in de sprint wordt opgepakt. Er wordt dan niet gekeken of deze nou ready is of niet. Dit komt omdat de Backlog gewoon leeg is.” Case 7*

*“Maar op het moment dat je de Definition of Ready en Done moet opstellen komt deze vol met allemaal eervolle dingen die je moet doen die er in de praktijk vaak niet van komen.” Case 7*

In de Definition of Ready waren een aantal overeenkomsten te vinden. Hieruit blijkt dat er meestal een aantal vaste onderdelen plaatsvinden tijdens het ready maken van User Stories. Dit zijn:

- Geprioriteerd
- Ingeschat
- Gereviewd
- How to test is opgesteld
- Klein genoeg/opgesplitst
- Voor iedereen duidelijk

Daarnaast kan het nog specifiekere (ontwerp)elementen bevatten zoals:

- Het datamodel is bijgewerkt aan de hand van de User Story;
- Er bestaat een schermontwerp bij de User Story;

In bijlage 5 is een aantal Definitions of Ready van de Cases te vinden.

### **Flow naar ready**

De Flow naar ready werd vaak inzichtelijk gemaakt. User stories verkeerde dan in een bepaalde staat voor deze de sprint in gingen. Dit werd vaak aangeduid met de status: 'Open for design' 'In design' 'Testable' en 'Ready'. 'In design' was in dit geval het moment waar de informatieanalist/ontwerper mee aan de slag ging. Deze had in een aantal gevallen weer zijn eigen flow waarin inzichtelijk werd gemaakt in welke status de User Story verkeerde. In de kolom Testable ging het om het testen van de User Story voldeed aan de Definition of Ready. Hier was dus plek voor de validatie. Het visueel maken van deze Flow werd als erg prettig ervaren aangezien iedereen kon zien wat er aan zat te komen. In één case hingen de Flow naar ready en het Scrumboard naast elkaar. Op deze manier zijn beide processen in één oogopslag duidelijk.

*“Wij houden ons niet echt aan de Definition of Ready want anders hadden we in het verleden heel vaak geen werk gehad. Dus dan kan je kiezen van ik hou me er aan en ik ga niks doen. Of we hanteren hem niet en we gaan aan de slag. Het gaat nu wat beter omdat we de voorbereiding iets beter in kaart hebben.” Case 2*

*“Maar ik moet bekennen dat we ons daar niet altijd goed aan hebben gehouden en hetzelfde geldt voor de Definition of Done.” Case 1*

*“Het heeft heel lang geduurd voordat er ready Stories de sprint in gingen.” Case 5*

### 3.1.3 De rol van de informatieanalist

De rol van de informatieanalist was bij iedere case ingevuld. In vijf van de acht gevallen was de informatieanalist tevens de functioneel ontwerper. Van onderscheid in deze rol was dan geen sprake. De rol van de informatieanalist bestond in dit geval uit het opstellen, specificeren en uitwerken van User Stories. Op deze manier werd de gewenste input geleverd voor de sprint.

In de drie cases waar de rol van informatieanalist en de functioneel ontwerper wel gescheiden waren, was er sprake van grote organisaties met een erg duidelijke hiërarchie. De informatieanalist was te allen tijden bedoeld om de Product Owner te ondersteunen. In twee gevallen bleek de informatieanalist veel taken van de Product Owner te hebben. In deze gevallen werd er aangegeven dat dit geen juiste rol voor de informatieanalist is. Er werd aangegeven de Product Owner verantwoordelijk moet zijn en dat de informatieanalist slechts ondersteunt. Belangenverstrengeling was iets wat in deze gevallen werd genoemd aangezien een informatieanalist in de meeste gevallen zal denken vanuit ICT en minder vanuit de Business.

De plek van de informatieanalist was in de meeste gevallen op de rand. Meestal (in 6 van de 8 gevallen) zat hij in een eigen team. De input van de informatieanalist kwam in alle gevallen vanuit de business. De informatieanalist zorgde ervoor dat aan de hand van deze informatie Epic's of User stories werden beschreven.

Ondanks dat de rol van de Product Owner niet specifiek een topic was, is deze rol bij vrijwel iedere case aan bod gekomen. Dit kwam in bijna alle gevallen doordat er veel problemen waren rondom de invulling van deze rol. De oorzaken hiervan waren vaak: te weinig mandaat, te weinig tijd, te weinig eigenaarschap of te weinig kennis.

*“Nee dat zijn wij zelf Functioneel Ontwerp bestaat niet echt meer hé.” Case 8*

*“Nee wat dat betreft zit ik ook echt op de rand. Omdat je als analist ook eigenlijk onderdeel van het team bent pak je ook andere dingen op.” Case 1*

*“Ja dat klopt en dan krijg je belangenverstrengelingen. Dus ik denk zeker dat op het gebied van analyse en ontwerp in het voortraject het heel erg belangrijk is om een goede Product Owner te benoemen. Eentje die de rol ook echt kan uitvoeren.” Case 6*

*“Dan moeten de User Stories nog bedacht worden. Dat wordt hier sowieso niet door de Product Owner gedaan. Vanuit het team wordt die Backlog eigenlijk gebouwd.” Case 6*

*“Dat is ook echt wel een voorwaarde. De Product Owner moet eigenlijk twee rollen hebben. Aan de ene kant moet hij het mandaat hebben. Dat is eigenlijk best een hoge functie. Aan de andere kant moet hij redelijk inhoudelijk en hands on de Product Backlog vullen en bijhouden. Ik denk vooral dat je die in grote organisaties moeilijk gaat vinden. Wat je daar altijd zal zien is dat er een informatieanalist bij is.” Case 1*

# Hoofdstuk 4

## Conclusie

*In paragraaf 4.1 staat de conclusie over de verkregen resultaten vanuit de casestudy beschreven. In deze conclusie wordt er dieper ingegaan op de resultaten uit het voorgaand hoofdstuk en wordt er gereflecteerd richting het referentiekader. In paragraaf 4.2 wordt puntsgewijs samengevat wat de conclusie concreet betekent voor het advies. Paragraaf 4.1 en 4.2 zijn beide beschreven aan de hand van de hoofdthema's. In paragraaf 4.3 volgt een conclusie over het Scrumproces in relatie tot de bevindingen.*



## 4.1 Conclusie resultaten casestudy

### 4.1.1 Komen tot de Product Backlog

#### Opstellen

Ondanks de verschillen blijkt er uit de analyse van de resultaten wel een constante factor. Dit is het benoemen en doorlopen van de processen waarvoor de IT-ondersteuning moet komen. Dit wordt gedaan met stakeholders en op deze manier komt de Product Backlog tot stand. Het blijkt erg belangrijk om dit high level te doen en voor het meest essentiële.

Van een echt concrete aanpak zoals deze wel staat beschreven in het referentiekader was in de meeste gevallen geen sprake omdat men hier nog naar op zoek is. Dit is niet erg vreemd want vaak werken organisaties nog niet heel erg lang volgens de Scrummethodiek en wordt Scrum vooral toegepast tijdens de realisatie. Hierdoor komt de Product Backlog nog vaak vrij ad-hoc tot stand of juist ver van te voren, op hoog niveau naar aanleiding van traditionele informatieanalyse(documenten).

De stappen zoals deze staan beschreven in het referentiekader is de aanpak die Quintor zelf hanteert om te komen tot de Product Backlog. Uit de casestudy blijkt dat het een aanpak is die erg aansluit bij de werkwijze die organisaties gewend zijn. Dit komt omdat de stappen die worden doorlopen overeenkomen met de elementen die traditioneel in een Use Case worden beschreven. Uit de literatuur is al eerder gebleken dat de Use Case een waardevol middel kan zijn om requirements mee te verkrijgen en vast te leggen. Door delen van de Use Case onderdeel te laten zijn van een interactieve sessie wordt er gebruik gemaakt van een sterk element uit de Use Case, namelijk: het beschrijven van requirements (User Stories) in een bepaalde context. (Leffingwell, 2012) Door processen te doorlopen met stakeholders en aan de hand daarvan User Stories te benoemen wordt deze context gecreëerd. Uit deze bevindingen kan worden geconcludeerd dat de aanpak die Quintor heeft, veel kan betekenen voor organisaties die deze stappen nog niet zo helder voor ogen hebben. Tevens is het belangrijk dat deze stappen zich herhalen gedurende het project aangezien de Product Backlog aan de hand van feedback van stakeholders en nieuwe inzichten evolueert. (Pichler, 2012)

#### Visueel artefact

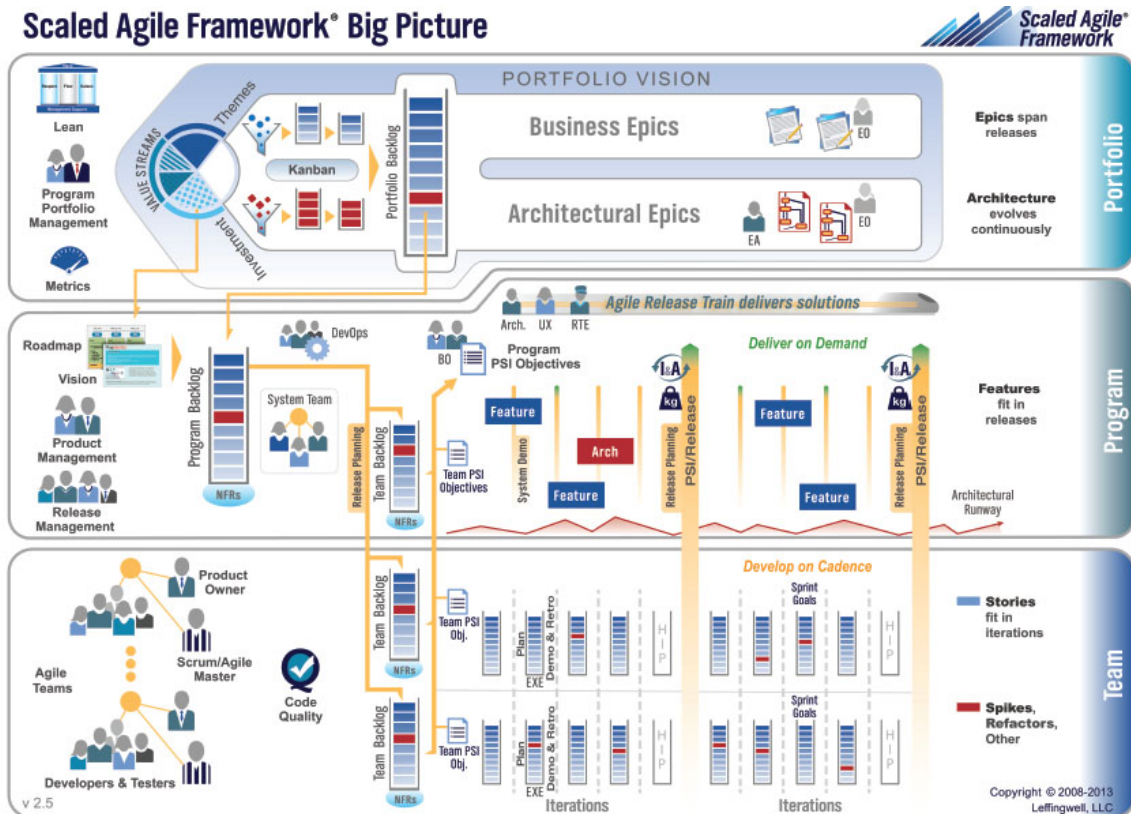
Uit de casestudy bleek dat het verstandig is om deze processen niet alleen te benoemen maar ook visueel te maken. Dit bleek een ondersteuning tijdens het vormen van de (initiële) Product Backlog te zijn maar wordt ook als een welkome aanvulling tijdens het project gezien. Op deze manier blijft namelijk de context bestaan gedurende het project. Om deze reden kan het verstandig zijn om ook gedurende het project het visueel artefact up-to-date te houden. Zo wordt het uiteindelijke doel inzichtelijk en kan men de voortgang tonen. Het creëert een gemeenschappelijk taal en nodigt mensen uit om feedback te leveren. Dit geeft vertrouwen. Doordat het vaak geen officieel document is geeft het veel vrijheid aan het team. Een visueel artefact blijkt dus een waardevolle aanvulling op het referentiekader.

Het hebben van een visie was al in het referentiekader opgenomen als onderdeel van de Quintor aanpak om te komen tot de Product Backlog. Tijdens de casestudy is het hebben van een visie sterk benadrukt.

Vaak was deze visie te beperkt. Het hebben van een visie is een erg waardevol middel. Wanneer deze goed is opgesteld kan er voor een groot deel de Product Backlog uit worden gedestilleerd. Er zijn verschillende manieren waarop een visie binnen Agile/Scrum kan worden opgesteld. Een goede 'lichte' manier is bijvoorbeeld het creëren van een visie aan de hand van een prototype, mock-ups, scenario's en persona's. (Pichler, 2012)

### **Verschillende niveaus**

In een aantal gevallen is er aangegeven dat het opstellen van de Product Backlog vooral gebeurt aan de hand van de Watervalmethode en dat eigenlijk alleen de realisatie volgens Scrum gebeurt. In dit geval werd er ook wel gesproken over Water-Scrum-Fall. Deze term lijkt goed weer te geven wat er gebeurt wanneer organisaties gaan werken volgens Scrum. Daves West (2011) schreef in een onderzoeksrapport, waarin hij onderzoek deed naar Scrum in grote organisaties, dat Water-Scrum-Fall vaak de realiteit is voor organisaties wanneer deze gaan werken volgens de Scrummethodiek. Hij geeft in zijn artikel, *Water-Scrum-Fall Is The Reality Of Agile For Most Organizations Today*, aan dat dit niet perse negatief hoeft te zijn maar dat dit wel goed ingericht moet worden om te kunnen waarborgen dat de voordelen die met Agile behaald worden gewaarborgd blijven. Uiteindelijk moet er geprobeerd worden om het Water-gedeelte zo beperkt mogelijk te houden. Wat er anders gebeurt is dat men niet snel toekomt aan Just-Enough en Just-in-Time analyse en ontwerp. Het is dan namelijk niet meer Just-in-Time. Hierdoor wordt er vaak (ten onrechte) veel tijd in de analyse gestoken. De behandelde case waarin de Product Backlog op verschillende niveaus tot stand kwam lijkt een erg geschikte aanpak om te zorgen dat, ook bij grote projecten, het Water- (en Fall-gedeelte) zo licht mogelijk wordt gehouden. Op deze manier wordt ervoor gezorgd dat Scrum niet alleen tijdens de realisatie wordt toegepast maar dat de hele organisatie meer Agile wordt. Als ook op hoog niveau de organisatie zo in elkaar zit wordt Just-in-Time en Just-Enough een stuk makkelijker. Deze aanpak vertoont veel overeenkomst met het Scaled Agile Framework van Dean Leffingwell (2011) figuur 1.



Figuur 1 Scaled Agile Framework Bron: <http://scaledagileframework.com/> geraadpleegd op 25 november 2013

### Scaled Agile Framework

Dit Framework beschrijft een aanpak om de hele organisatie zo in te richten dat alles op elkaar aansluit en verschillende lagen in de organisatie worden meegenomen in het ontwikkeltraject. Het Scaled Agile Framework is gebaseerd op zowel Agile als Lean principes. Lean richt zich in tegenstelling tot Agile meer op de hele organisatie. Lean richt zich hoofdzakelijk op de 'value stream' en gaat uit van verschillende gedachten en principes die de 'time to market' verkorten en waste en vertragingen tegen gaan. Het Scaled Agile Framework richt zich op de big-picture waarin de gehele Enterprise in oenschouw wordt genomen. Het model definieert drie verschillende niveaus namelijk:

- Het teamniveau
- Het programmaniveau
- Het portfolioniveau

Per niveau wordt tevens onderscheid gemaakt tussen de staat van de requirements.

- User stories: op team niveau zijn de requirements uitgewerkt in de vorm van User Stories.
- Features: bestaan op programmaniveau en zijn kleiner dan Epic's maar nog niet klein genoeg om als User Story in een sprint te worden opgenomen. Features worden gebruikt om het gat tussen User stories en scenario's te overbruggen.
- Epics: worden op portfolioniveau geformuleerd en zijn een beschrijving van een bepaalde visie of een scenario.

Op deze manier zijn er drie verschillende Product Backlogs op drie verschillende niveaus. Alle Product Backlogs zijn geprioriteerd en worden overgedragen aan een lager niveau zodat er uiteindelijk op teamniveau gerealiseerd kan worden aan de hand van User Stories.. (Leffingwell, 2011)

### **Bottom-up of top-down**

Er zijn dus eigenlijk twee manieren om te komen tot de Product Backlog. Ten eerste door middel van één of meerdere interactieve sessies waarin high level het proces wordt doorlopen en User Stories worden benoemd. Ten tweede op een manier waarbij de eerste aanzet van de Product Backlog al op portfolioniveau wordt gerealiseerd op basis van Epic's en dat op deze manier Product Backlogs op verschillende niveaus in de organisatie in elkaar over lopen. De eerste aanpak sluit vooral goed aan wanneer Scrum bottom-up wordt geïmplementeerd. De tweede aanpak is een manier om top-down met Scrum bezig te gaan. Vaak is de tweede aanpak een gevolg nadat tijdens de eerste aanpak Scrum wordt ingevoerd en er later ook aansluiting wordt gezocht op een hoger niveau

### **Insteken op portfolioniveau**

Deze thesis steekt vooral in op User Story niveau en dus een bottom-up aanpak. Als men de big-bigpicture wil zien zoals in de Scaled Agile Framework en de behandelde case, betekent dit dat alle lagen van de organisatie moeten worden meegenomen. Omdat deze thesis oorspronkelijk niet gericht was op een soortgelijke aanpak, maar omdat dit tijdens deze conclusie fase wel een relevant onderdeel bleek te zijn is de onderzoeker opzoek gegaan naar nog een organisatie die op deze manier werkt. In een informeel ongestructureerd interview is Albert Alma van AEGON benaderd. Ook hij heeft een aanpak ontwikkeld die uitgaat van Scrum in de realisatie en waar de rest van de organisatie in wordt meegenomen. Toen hij werd geconfronteerd met de vraag of hij dit had gebaseerd op het Scaled Agile Framework bleek dat hij dit in eerste instantie niet kende, maar er later achter was gekomen dat iets soortgelijks al bestond. Dit geeft aan dat organisaties zelf bezig zijn om het portfolioniveau aan te laten sluiten bij de uiteindelijke realisatie waardoor de gehele organisatie meer Agile wordt in plaats van dat dit zich beperkt tot de realisatie. Vanzelfsprekend wordt er op deze manier ook een bepaalde richting geven aan analyse en ontwerp aangezien het framework een manier beschrijft waarop je in diverse stadia op diverse niveaus binnen een organisatie om kunt gaan met analyse en ontwerp

### 4.1.2 Ready maken

Er kan worden geconcludeerd dat er verschillende opvattingen bestaan over wat ready betekent. Soms had dit alleen betrekking op het formuleren van een User Story en in een andere gevallen betekende het dat er een Use Case was opgesteld. Aan de hand van de analyse van de resultaten blijken er wel een aantal vaste elementen te zijn rondom dit begrip namelijk:

- Geprioriteerd
- Ingeschat
- Gereviewd
- How to test is opgesteld
- Klein genoeg/opgesplitst
- Voor iedereen duidelijk

De belangrijkste twee die veel zijn genoemd tijdens de Casestudy waren:

1. Een User Story moet voor iedereen duidelijk zijn.
2. Een User Story moet kunnen worden ingeschat tijdens de Sprint Planning Meeting.

Deze twee punten hangen erg samen omdat een User Story over het algemeen niet kan worden ingeschat wanneer deze niet duidelijk is. Ondanks deze overeenkomsten bleek verder dat hoe, wanneer, door wie en in welke mate het ready maken verder gebeurde sterk varieerde. Het ready maken blijkt een proces waarin voldoende werkvoorraad creëren een groot knelpunt is. In het referentiekader staat dat men moet proberen 1 à 2 sprints voor te lopen. In alle cases was dit ongeveer ook het streven. Dit bleek in de meeste gevallen niet haalbaar. Hieruit blijkt dat Just-in-Time en Just-Enough analyse erg lastig is voor organisaties.

#### Specificatie

Het blijkt dat wat er het team in gaat, sterk afhangt van de informatiebehoefte van het team. Een ontwikkelteam dat aan weinig voldoende heeft en dat zelf in staat is om (een deel van) de oplossing te bepalen en gaten op te vullen in analyse en ontwerp wordt als succes ervaren. Vaak gaat dit samen met veel werkervaring en ervaring met iteratieve Scrumtrajecten. Het tegenovergestelde hiervan is dat in sommige teams een grote informatiebehoefte is en dat hierdoor het ready maken veel omvangrijker wordt. De meest voor de hand liggende oorzaak voor een hoge informatiebehoefte lijkt een bepaald kennisniveau te zijn. Andere factoren voor de omvangrijkheid van het ready maken zijn: kennis in de organisatie, afhankelijkheden van andere systemen, een bepaalde cultuur, geen

*“Ze hadden gewoon de kennis en de kunde niet om dat soort gegevens voor de ontwikkelaars te kunnen opleveren. Ze snapte ook wel dat wij het niet zelf konden verzinnen maar dat is heel lang een probleem geweest en dan ook nog de vraag van hoe gaan we dat opschrijven.” Case 5*

*“De kennis binnen de organisatie was daar ook heel laag van en heel moeilijk te krijgen. Eigenlijk had ik daar gewoon een expert van in het team willen hebben voor een sprint om dat op te lossen maar dat krijg je gewoon niet want die mensen waren gewoon niet beschikbaar.” Case 1*

goede Product Owner en gescheiden locaties. Dit blijken factoren waar binnen het referentiekader in eerste instantie geen aandacht aan was besteed, omdat veel van deze factoren buiten analyse en ontwerp lijken te vallen. Echter deze factoren werden veel genoemd tijdens de casestudy en blijken van grote invloed op het Scrumproces en dus ook op analyse en ontwerp. Het is daarom belangrijk om hier rekening mee te houden in projecten.

Om met voldoende input voor het team te komen wordt de Use Case of elementen uit de Use Case gebruikt. Dit lijkt een geschikt formaat aangezien iedereen het goed kent. Omdat er moet worden gestreefd naar Just-Enough specificatie moet het gebruiken van Use Cases zo licht mogelijk gehouden worden. Dit kan worden bereikt door veel afstemming te hebben met stakeholders om op die manier Just-in-Time, Just-Enough, te beschrijven. Er kan worden geconcludeerd dat dit valt of staat met het hebben van een goede informatieanalist. Hierbij is belangrijk dat deze in staat is om steeds te bepalen wat het meest essentiële is dat nodig is voor het ontwikkelteam.

### **Validatie**

Het blijkt dat valideren, of iets ready is, belangrijk is en veel gebeurt. Dit dient te gebeuren aan de hand van de Definition of Ready. Opvallend was dat ondanks de validatiemomenten er toch veel items de sprint in gingen die niet ready waren. Dit geeft aan dat valideren wel een bepaalde richting geeft, maar geen zekerheid. Hieruit kan worden geconcludeerd, dat ruimte voor specificatie tijdens de sprint echt een eis is ook wanneer er wordt gekozen voor een losstaand ready-team. De uitgebreide reviews lijken dan voornamelijk ook te ontstaan om te kunnen waarborgen dat alles op voorhand juist gespecificeerd is. Dit wordt geconcludeerd, omdat de meest uitgebreide reviewprocedures aanwezig waren bij cases, waarin sprake was van een losstaand ready-team. Het ready maken voorafgaand aan een sprint lijkt noodzakelijk, maar ook een valkuil, om teveel terug te gaan richting de Watervalmethode. Het is belangrijk om hier altijd voor te blijven waken. Het is namelijk niet raar dat dit snel zal gebeuren. De Watervalmethode sluit erg goed aan bij hoe de mens van nature denkt en de wens heeft om bepaalde dingen te kunnen voorspellen en te plannen. Dit is ook de reden waarom de Watervalmethode erg populair is geweest en nog steeds wordt gebruikt. (Leffingwell, 2012)

### **Definition of Ready**

Het is denkbaar dat, als de Definition of Ready wel goed wordt gehandhaafd en realistisch is, er minder vaak blijkt dat items onvoldoende ready zijn. In een case waar de Definition of Ready op deze manier werd gebruikt, was er geen knelpunt rond het hebben van niet ready items tijdens de sprint. Het is belangrijk om ook gedurende de sprints te beoordelen of de Definition of Ready niet te hoge eisen stelt. Te hoge eisen zorgen ervoor dat er teveel wordt gedaan dan op dat moment zinvol is. Op deze manier wordt Just-in-Time en Just-Enough-analyse-en-ontwerp tegengegaan.

Er moet sprake zijn van een erg flexibele informatieanalist die niet alleen als taak ziet User Stories ready te maken maar ook mee werkt aan het doel van de sprint. Dit is het gevaar wanneer ready maken als een parallelle activiteit wordt uitgevoerd. Zelfs wanneer zo'n scheiding niet fysiek is, kan het wel zo aanvoelen blijkt uit de casestudy. Dit vraagt veel van het vermogen van de informatieanalist.

### **Documentatie**

Het opleveren van documentatie blijkt in een aantal gevallen onderdeel te zijn van de Definition of Ready. Echter bleek in veel gevallen de ontwikkelaar aan veel minder genoeg te hebben dan volledige specificatie en documentatie. Dit was zelfs het geval wanneer het ontwikkelteam een relatief hoge informatiebehoefte had. Het lijkt daarom niet verstandig om documentatie onderdeel van de Definition of Ready te maken. Documentatie kan daarom beter onderdeel zijn van de Definition of Done. Een functioneel ontwerper, die de vereiste documentatie vervaardigt, blijft op deze manier wel nodig tijdens de sprint. Daarnaast blijkt het mogelijk om door in een vroeg stadium kritisch te kijken naar de documentatie, deze kan worden beperkt tot een minimum. Ook voor documentatie geldt dus: Just-in-time en Just-Enough.

### **Flow naar ready**

Het inzichtelijk maken van de Flow naar ready gebeurde in bijna alle gevallen. De categorieën waarin de stories verkeerden waren uitgebreider dan die in het referentiekader stonden beschreven. De categorieën kunnen beter bestaan uit: 'Open for design' 'In design' 'Testable' en 'Ready'.

### 4.1.3 De rol van de informatieanalist

De informatieanalist zat vaak of meer richting de business of meer richting het team. Zoals in het referentiekader is aangegeven is de informatieanalist volgens de geraadpleegde theorieën op verschillende plaatsen denkbaar. De casestudy heeft echter uitgewezen dat bij complexere projecten en/of projecten met meerdere teams (iets dat vaak samen gaat), de plek van de informatieanalist het best past in een apart team dat zich bezighoudt met het ready maken van User Stories. In de meeste gevallen was dit ook zo ingericht. Zoals uit de vorige alinea blijkt, is het voor de informatieanalist(en) belangrijk om beschikbaar te blijven tijdens de sprint en zich betrokken te voelen bij het behalen van het sprintdoel. De plek van de informatieanalist is dan dus echt tussen het team en de Product Owner in. Bij kleinere projecten en minder complexe projecten kan de informatieanalist wel onderdeel zijn van een team. Het blijkt uit de casestudy dat het in dit geval veel makkelijker is om tijdens een sprint de specificatie te doen en User Stories ready te maken.

Uit de casestudy blijkt dat de informatieanalist snel de rol van de Product Owner aanneemt en dat dit niet als wenselijk wordt beschouwd. Dit is ook in geen enkel geval een bewuste keuze geweest maar is vooral het gevolg wanneer er geen goede Product Owner beschikbaar is. Zoals ook al stond beschreven in het referentiekader vertonen de rol van de informatieanalist en die van de Product Owner veel overlap. Door middel van de casestudy is bevestigd dat de rol van de informatieanalist echt ondersteunend dient te zijn en dus niet ter vervanging.

Tot slot kan er worden geconcludeerd dat een scheiding tussen informatieanalisten en functioneel ontwerpers binnen Scrum niet goed past met betrekking tot het ready maken van User Stories. In de cases waar deze scheiding er niet was, was men veel meer gefocust op zorgen dat er voldoende input kwam voor de sprint in plaats van de focus te leggen op documenten. Zoals in de vorige alinea is aangegeven blijft de rol van een Functioneel ontwerper wel denkbaar binnen Scrum maar dan vooral tijdens de sprint voor bijvoorbeeld het opstellen van documentatie conform de Definition of Done. Tevens kunnen vragen worden beantwoord uit het ontwikkelteam die tijdens de sprint ontstaan voordat er wordt teruggekoppeld richting de informatieanalist.



## 4.2 Advies in relatie tot de bevindingen

In deze paragraaf zijn de belangrijkste punten uit de conclusie vertaald naar adviserende richtlijnen. Dit is gedaan per behandeld onderdeel uit de conclusie. De conclusie is gereflecteerd richting het eerder opgestelde referentiekader. Hieruit blijkt dat de punten die eerder zijn opgesteld vanuit de theorie goed aansluiten bij de bevindingen vanuit de casestudy. De punten die in dit hoofdstuk staan beschreven zullen daarom samen met het referentiekader onderdeel zijn van de aanbevelingen aan Quintor.

### 4.2.1 Komen tot de Product Backlog

#### Opstellen

- Het opstellen van de Product Backlog dient in eerste instantie te gebeuren aan de hand van de meest essentiële functionaliteit.
- Het creëren van een bepaalde context tijdens het opstellen van de Product Backlog is essentieel.

#### Visueel artefact

- Er kan een visueel artefact dat (high level) de processen van het te bouwen systeem weergeeft worden opgesteld in het voortraject/sprint 0. Op deze manier kan de context worden verduidelijkt en kunnen User Stories worden benoemd. Het visueel artefact creëert tevens een gemeenschappelijk taal waarmee voortgang kan worden getoond en vertrouwen van stakeholders kan worden gewonnen. Het hebben van een visueel artefact nodigt uit om fouten te benoemen waardoor er constant feedback wordt geleverd. Door het geen officiële status te geven ontstaat vrijheid
- Het visueel artefact kan up-to-date worden gehouden aan de hand van nieuwe inzichten zodat het een middel blijft gedurende het project.

#### Verschillende niveaus

- Wanneer er bij grote projecten en/of organisaties sprake is van het beschrijven van Epic's en men het portfolioniveau beter wilt laten aansluiten bij de realisatie volgens Scrum kan er worden overwogen om User Stories en teams op verschillende niveaus te beschouwen zoals in het Scaled Agile Framework wordt gedaan. Dit heeft als doel ervoor te zorgen dat er geen grote scheiding ontstaat tussen het beschrijven van Epic's op hoog niveau en de realisatie volgens Scrum.

### 4.2.2 Ready maken

#### Flow naar ready

- De User Stories dienen in verschillende categorieën worden ingedeeld om de fase waarin de User Story verkeert inzichtelijk te maken: 'Open for design' 'In design' 'Testable' en 'Ready'.

#### Validatie

- Checken of iets ready is kan gebeuren aan de hand van de Definition of Ready. Hoewel validatie noodzakelijk is geeft het niet altijd zekerheid omdat vaak, ondanks een validatie, User Stories niet ready blijken. Het is daarom belangrijk om dit onderdeel niet te zwaar te maken en tijd over te houden om tijdens de sprint problemen die ontstaan op te lossen.

### Definition of Ready

- Gedurende het project moet er worden gekeken of de Definition of Ready wel realistisch is en moet deze zo nodig worden bijgesteld. Op deze manier wordt Just-in-time en Just-Enough-analyse-en-ontwerp mogelijk.
- De Definition of Ready bevat in essentie vaak in ieder geval de volgende onderdelen:
  - Geprioriteerd
  - Inschatbaar
  - Gereviewd
  - How to test is opgesteld
  - Klein genoeg/opgesplitst
  - Voor iedereen duidelijk

### Specificatie

- Het uitbreiden van de User Story kan gebeuren aan de hand van elementen vanuit de Use Case. Hierbij is het belangrijk om te allen tijde naar de noodzaak te blijven kijken.
- Factoren als kennisniveau/informatiebehoefte van het team, kennis in de organisatie kennis die nodig is bij het realiseren van de IT-ondersteuning, afhankelijkheden van andere systemen, een bepaalde cultuur, geen goede Product Owner, documentatie-eisen en gescheiden locaties zorgen ervoor dat het ready maken omvangrijker wordt. Hier dient men op te acteren tijdens de Flow naar ready en bij aanvang van het project bij stil te staan.

### Documentatie

- In een voortraject is het belangrijk de documentatie-eis te beperken tot een minimum en hierbij vooral te kijken naar de noodzaak.
- Er dient te worden overwogen om documentatie op te nemen als onderdeel van de Definition of Done en niet van de Definition of Ready. Dit is verstandig omdat anders het gevaar bestaat dat er te vroeg, te veel, wordt gedaan. Vervolgens moet het dan later aangepast worden, omdat de documentatie dan (deels) niet meer aansluit bij hetgeen dat gebouwd is.

#### 4.2.3 De rol van de informatieanalist

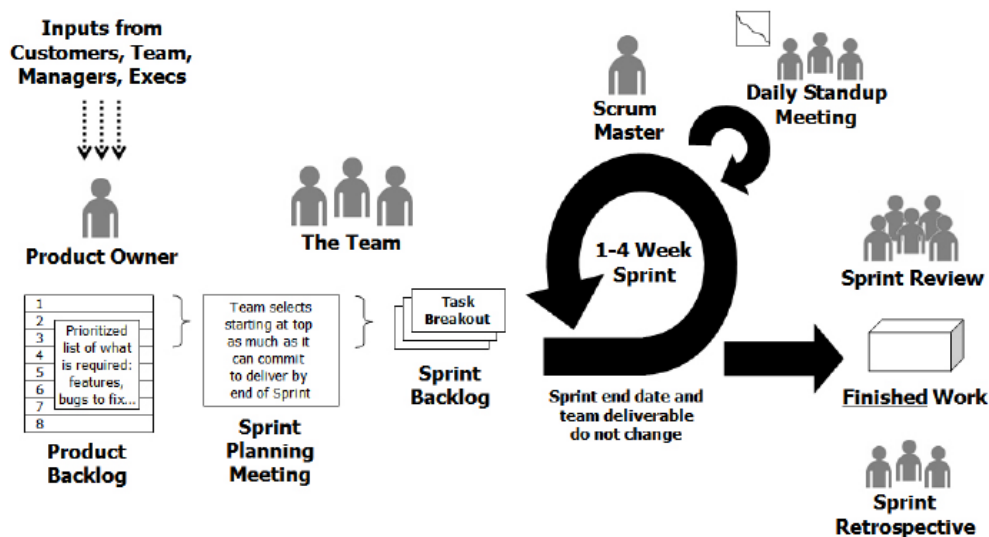
- De plek van de informatieanalist is tussen het team en de Product Owner in. Dat betekent dat deze zowel meewerkt in de sprint als in de Flow naar ready. Bij kleine projecten kan de plaats van de informatieanalist het beste in het team zijn.
- De informatieanalist is te allen tijde beschikbaar voor het team zodat deze betrokken blijft voor het beantwoorden van vragen en het behalen van het sprintdoel.
- Met betrekking tot het ready maken is het niet zinvol tussen informatieanalisten en functioneel ontwerpers een scheiding aan te brengen.
- Het is belangrijk om te waarborgen dat de informatieanalist puur ter ondersteuning van de Product Owner dient en niet ter vervanging. De Product Owner heeft specifieke belangen, kennis en mandaat en dient daarom de verantwoordelijkheid te hebben en te nemen.

### 4.3 Conclusie Scrumproces

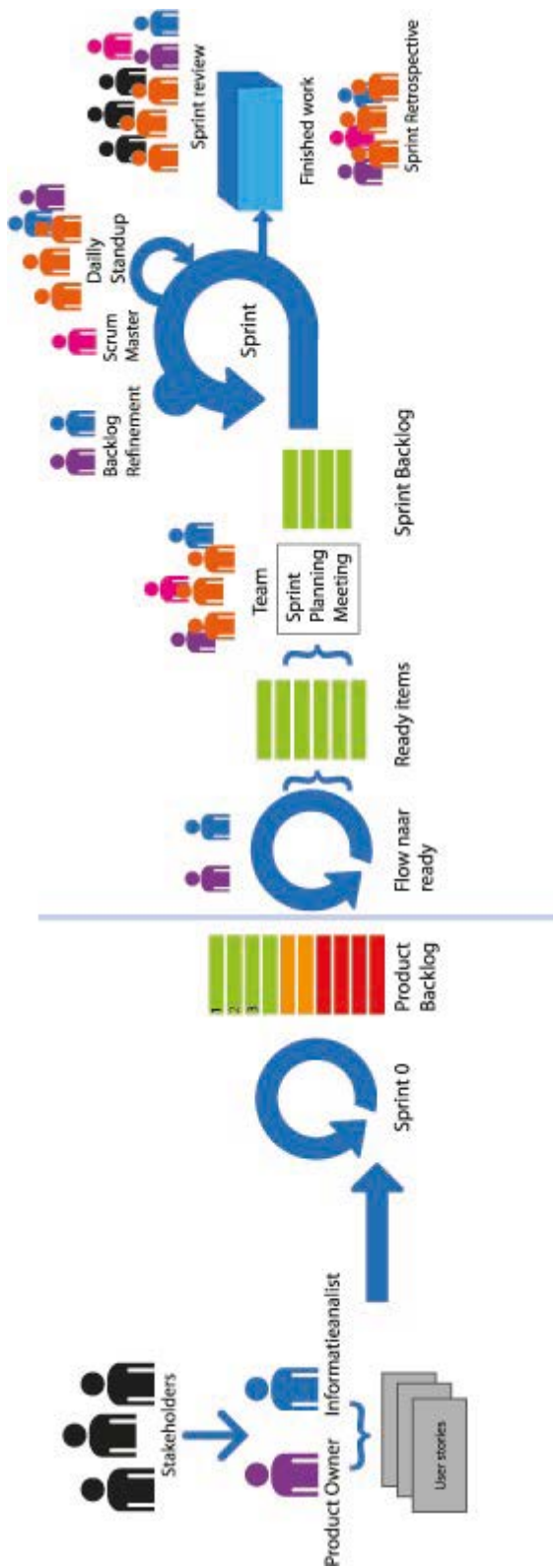
Wanneer we nu naar het Scrumproces kijken zoals deze wordt beschreven door grondleggers van Scrum, zoals Jeff Sutherland, kan worden geconcludeerd dat een aantal essentiële onderdelen die tijdens het theoretisch kader, het vooronderzoek en de casestudy naar boven zijn gekomen ontbreken. Het Scrumproces, zoals Sutherland & Schwaber (2007) deze weergegeven in *The Scrum Papers* is te zien in figuur 2. In figuur 3 wordt het aangepaste proces weergegeven. Hierin worden de onderdelen weergegeven die essentieel zijn gebleken op basis van het onderzoek.

#### Toelichting op de afbeelding

In dit model wordt de Product Owner rol ondersteund door een informatieanalist maar zoals eerder is aangegeven is deze rol ook als onderdeel van het team denkbaar. Het gaat er in het figuur vooral om dat de rol een plek kent binnen Scrum. Tevens wordt in figuur 2 sprint 0 weergegeven en de Flow naar ready. Sprint 0 staat hier voor de lijn omdat deze eenmalig voorafgaand aan de eerste sprint gebeurt. De uitkomst van Sprint 0 is een geprioriteerde Product Backlog met daarin User Stories die voorzien zijn van een Definition of Ready en een Definition of Done. Sprint 0 is in dit model het moment dat de visie wordt/worden opgesteld, interactieve sessie(s) wordt/worden gehouden en visueel(e) artefact(en) worden opgesteld. Vervolgens wordt de User Story ready gemaakt tijdens de Flow naar ready. Niet alle Stories worden ready gemaakt maar alleen de hoogst geprioriteerde. Er worden in ieder geval genoeg items Stories gemaakt voor 1 à 2 sprints. De Product Owner is hier verantwoordelijk voor en wordt daarbij ondersteund door de informatieanalist. Tijdens de Sprint is in het Scrumproces de Flow naar ready een parallelle activiteit om steeds weer hoog geprioriteerde items ready te maken voor de volgende sprint.



Figuur 2 Scrumproces bron: [http://scruminc.com/tl\\_files/scrum\\_inc/documents/ScrumPapers.pdf](http://scruminc.com/tl_files/scrum_inc/documents/ScrumPapers.pdf) geraadpleegd op 6 september 2013



Figuur 3 aangepast Scrumproces

# Hoofdstuk 5

## Aanbevelingen

*Op basis van de resultaten en de conclusie kunnen aanbevelingen aan Quintor worden gedaan. In paragraaf 5.1 staat het advies op basis van de resultaten van het onderzoek beschreven. In paragraaf 5.2 worden er aanbevelingen gedaan voor mogelijk vervolgonderzoek. In dit hoofdstuk wordt antwoord gegeven op de laatste centrale vraag.*

*Wat leert ons de beoordeling van de analyseresultaten met het oog op het doen van aanbevelingen op het gebied van analyse en ontwerp in een Scrumproject?*

- a. Op welke manier kunnen deze inzichten bijdragen aan een mogelijke oplossing?*
- b. Welke aanbevelingen kunnen er worden gedaan op basis van het onderzoek?*

## 5.1 Aanbevelingen Analyse en Ontwerp

### Richtlijnen

Door het samenvoegen van de bevindingen uit de literatuur, het vooronderzoek (paragraaf 2.2) en de casestudy (paragraaf 4.2) zijn er adviserende richtlijnen ontstaan. De aanbeveling aan Quintor is om deze richtlijnen in acht te nemen tijdens het uitvoeren van projecten volgens de Scrummethodiek. De richtlijnen zijn met name van belang voor projecten waarin:

- Er wordt gezocht naar een manier om de Product Backlog te vormen.
- Er onduidelijkheden zijn rondom het ready maken van User Stories.
- Er een informatieanalist betrokken is.

Het overzicht van de richtlijnen is te vinden in bijlage 6. Doordat de richtlijnen gebaseerd zijn op dat wat er in de praktijk en speelt en wat in de theorie beschreven staat wordt verwacht dat de deze een directe verbetering zijn op het gebied van analyse en ontwerp in Scrumprojecten. Deze richtlijnen geven een handvat om het gat dat ontstaat op het gebied van analyse en ontwerp op te vullen. Hiermee is de doelstelling van het project behaald en wordt de vraag van de opdrachtgever beantwoord. De richtlijnen zijn ingedeeld aan de hand van de behandelde thema's. Er moet worden benadrukt dat het richtlijnen zijn en geen regels. Uit de conclusie is immers gebleken dat projecten erg kunnen verschillen en daarom dus ook de aanpak. Er is daarom voor gekozen om de richtlijnen een bepaalde vorm van abstractie mee te geven zodat deze naar eigen behoeften kunnen worden ingezet. Voor alles geldt dan ook: pas het toe wanneer dit relevant en vooral ook nuttig is. Tot slot wordt er aanbevolen om Figuur 3 uit het vorig hoofdstuk te gebruiken bij Quintor om het 'aangepaste' Scrumproces visueel weer te geven.

### Interactieve sessies

Uit het onderzoek blijkt Quintor al over een concrete aanpak te beschikken. Dit geldt met name voor het komen tot de Product Backlog door middel van interactieve sessies. De volgende aanbevelingen kunnen aan Quintor worden gedaan met betrekking tot deze aanpak:

1. Quintor doet er verstandig aan deze aanpak te gaan verspreiden. Op deze manier kunnen zij andere organisaties ondersteunen en opleiden. Dit is tevens een manier op meer marktgebied te verwerven. Om dit te doen kan Quintor trainingen geven waarbij men leert werken volgens deze aanpak.
2. Quintor dient zich hierbij te verdiepen in de aanvullende punten die vanuit de theorie en de casestudy naar voren zijn gekomen en te onderzoeken hoe zij zich deze punten eigen kunnen maken.
3. Quintor dient hiermee intern te beginnen. Tijdens de casestudy is gebleken dat niet alle medewerkers van Quintor deze aanpak helder voor ogen hebben. Daarom lijkt het een eerste logische stap om hier te beginnen.

## 5.2 Aanbevelingen vervolgonderzoek

De eerste aanbeveling voor een vervolgonderzoek heeft betrekking op het insteken op portfolioniveau vanuit Quintor. Er wordt verwacht dat het Scaled Agile Framework hiervoor een goed uitgangspunt is. Hiervoor kan worden onderzocht wat het Scaled Agile Framework concreet voor Quintor kan betekenen. De bevindingen uit dit onderzoek kunnen hiervoor het startpunt vormen. De twee organisaties die zijn behandeld in dit onderzoek, die een soortgelijke aanpak hanteren, kunnen als casestudy worden gebruikt. De insteek moet tijdens dit vervolgonderzoek, in tegenstelling tot dit onderzoek, liggen op de big-picture en niet op één specifiek gebied. Het is belangrijk om te vermelden dat bovenbeschreven aanbevelingen niet het één of het ander betekent. Deze twee aanpakken kunnen best naast elkaar bestaan of zelfs samengaan. Het insteken op portfolio niveau is vooral relevant in grote hiërarchische organisaties waar sprake is van Enterprise softwareontwikkeling.

De laatste aanbeveling heeft betrekking op de rol van de Product Owner. De rol van de Product Owner is in vrijwel iedere case als knelpunt benoemd maar heeft niet de focus van dit onderzoek gehad. Daarom lijkt een onderzoek naar specifiek de rol van de Product Owner en hoe die effectief ingevuld kan worden zinvol te zijn.

# Hoofdstuk 6

## Discussie

*Dit hoofdstuk betreft de afsluiting van deze thesis. In dit hoofdstuk wordt er in paragraaf 6.1 ingegaan op de resultaten en de gebruikte methode. In paragraaf 6.2 is er aandacht voor beperkingen die mogelijk het resultaat hebben beïnvloed. Gemaakte keuzes van de onderzoeker worden hier toegelicht.*



## 6.1 Resultaten

In de oorspronkelijke onderzoeksopzet lag de nadruk op het ontdekken van aanvullingen op de Scrummethodiek, op het gebied van analyse en ontwerp, door de focus te leggen op knelpunten en succesfactoren tijdens de interviews. Echter bleek het referentiekader op bepaalde gebieden veel verder te zijn dan de behandelde projecten. Dit bleek uit het feit dat er hoofdzakelijk veel knelpunten werden genoemd. Dit kwam omdat er naast analyse en ontwerp vaak nog veel meer zaken niet helemaal lekker liepen in het Scrumproces. Hierdoor zijn veel elementen uit het referentiekader vooral bevestigd in plaats van aangevuld. Het doel is wel behaald maar het vooronderzoek en het theoretisch kader hadden op deze manier een groter aandeel dan in eerste instantie was verwacht.

Tijdens het vooronderzoek bleek er al een concrete aanpak bij Quintor te liggen om te komen tot de Product Backlog. Deze aanpak kwam overeen met wat er in de theorie stond beschreven maar bij Quintor was deze aanpak geleerd en ontwikkeld vanuit de behoefte in de praktijk. Daarom is ervoor gekozen om deze aanpak op te nemen in het referentiekader. Op deze manier is deze aanpak getoetst aan de bevindingen in plaats van om aan de hand van de bevindingen een aanpak te formuleren. Met betrekking tot het ready maken van User Stories en de rol van de informatieanalist is er meer aanvullende informatie verkregen. De diversiteit van de behandelde cases heeft er toe geleid dat er een breed beeld is verkregen van de werkelijkheid. Echter bleek door de diversiteit en hoeveelheid het wel lastig om deze resultaten te analyseren. Uiteindelijk bleek dit niet slecht of verkeerd maar eerder een conclusie. Namelijk dat ieder project verschilt en dat daarom niet één specifieke methode leidend hoeft te zijn voor een goede aanvulling op Scrum. Er is daarom voor gekozen om met het vormen van de aanbevelingen hier rekening mee te houden.

## 6.2 Beperkingen

Tijdens het vormen van het theoretisch kader bleek er veel literatuur te bestaan over Agile en Scrum. Omdat het onderwerp dat deze thesis behandelt een nieuw onderwerp is, was het tijdens de literatuurstudie lastig om echt concrete literatuur te vinden. Op internet was er daarentegen zeer veel te vinden over dit onderwerp. Hierdoor is er voor gekozen om ook andere bronnen te raadplegen zoals blogs. Oorspronkelijk stond er in de onderzoeksopzet dat er zowel iemand van Quintor zou worden geïnterviewd als iemand van de organisatie zelf. Dit was zo bedacht om te waarborgen dat niet uitsluitend de Quintor-visie naar voren zou komen. Het bleek helaas lastig om dit ook daadwerkelijk voor elkaar te krijgen in verband met volle agenda's. Tevens bleek dat de respondenten vanuit Quintor erg goed vanuit de organisatie konden spreken. Er is daarom voor gekozen om tijdens de analyse geen onderscheid te maken tussen interviews met medewerkers van Quintor en medewerkers vanuit de organisaties. Er zijn in totaal 10 interviews gehouden in plaats van 14 van de in totaal 8 behandelde cases.

In de oorspronkelijke onderzoeksopzet stond beschreven dat iedere organisatie als case behandeld werd. Echter bleek dat aanpakken per project zo erg verschilden dat er voor gekozen is om ieder project als case te beschouwen in plaats van iedere organisatie.

Zoals aan het begin van dit hoofdstuk is aangegeven, bleken veel organisaties nog niet helemaal volwassen te zijn in het gebruik van de Scrummethodiek. Dit bleek uit belemmeringen binnen organisaties en projecten die van invloed zijn op de uitvoering van het proces. Voorbeelden hiervan zijn knelpunten op het gebied van het hanteren van de Definition of Ready, het hebben van multidisciplinaire teams en een goede Product Owner. Deze knelpunten hadden direct betrekking op het gebied van analyse en ontwerp. Hierdoor kan men zich afvragen hoeveel van de knelpunten over zouden blijven wanneer Scrum wel op een juiste manier werd uitgevoerd. Dit gegeven kan het resultaat enigszins hebben beïnvloed.

Zoals ook tijdens de conclusie is benoemt steekt dit onderzoek op een heel specifiek onderdeel van softwareontwikkeling in, namelijk; analyse en ontwerp. Tijdens dit onderzoek groeide het vermoeden dat er wellicht in sommige gevallen beter naar het gehele plaatje gekeken kan worden in plaats van naar één specifiek onderdeel. Dit vermoeden werd bevestigd tijdens het behandelen van de laatste case. Hierin werd een aanpak gehanteerd die meerdere lagen van de organisatie betrof bij de realisatie in tegenstelling tot de andere cases. Het behandelen van deze aanpak leek in eerste instantie niet relevant voor deze thesis. Echter bleek deze aanpak veel te betekenen voor de manier waarop analyse en ontwerp gedaan werd. Doordat het theoretisch kader en het referentiekader niet zijn ingestoken op dit niveau is dit onderwerp onderbelicht in dit onderzoek. De bevindingen zijn uiteraard wel opgenomen in het rapport.

# Literatuurlijst

Agile Introduction For Dummies – Part I (2008, 7 januari). website agileintro.wordpress. Geraadpleegd op 3 oktober 2013, van <http://agileintro.wordpress.com/2008/01/07/agile-introduction-for-dummies-%e2%80%93-part-i/>.

Aken, J. van & Andriessen, D. (2011). *Handboek ontwerpgericht wetenschappelijk onderzoek: Wetenschap met effect*. Den Haag: Boom Lemma.

Anders, A (2013). *Start a Scrum Project with Sprint 0*. Geraadpleegd op 3 oktober 2013, van <http://coding.abel.nu/2013/01/start-a-scrum-project-with-sprint-0/>.

Architech Solutions (2010). *Sprint Zero Where Your Software Project-Begins* Geraadpleegd op 3 oktober 2013, van <http://static.architech.ca/wp-content/uploads/2010/09/Sprint-Zero-Where-Your-Software-Project-Begins.pdf>.

Arendsen, M. e.a. (2008). *Succes met de requirements*. Den Haag : SDU uitgevers.

Beamont, S. (2009). *Flow to ready, iterate to done*. Geraadpleegd op 3 Oktober 2013 , van <http://blog.xebia.com/2009/07/04/flow-to-ready-iterate-to-done/>.

Cottmeyer, & Henson (z.d). *The Agile Business Analyst*. Geraadpleegd op 3 Oktober 2013 , van <http://www.agiledad.com/Documents/BAWhitepaperJune.pdf>.

Globale transcriptie( z.d.). Website notuleercentrum.nl. Geraadpleegd op 5 december, van [http://www.notuleercentrum.nl/transcriptieservice/globale\\_transcriptie.asp](http://www.notuleercentrum.nl/transcriptieservice/globale_transcriptie.asp).

Leffingwell, D. (2011). *Agile software requirements, lean requirements practices for teams, programs, and the enterprise*. Boston: Pearson Education, Inc.

Paetsch, F. e.a. (2003). Requirements Engineering and Agile Software Development. [elektronische versie] *Computer Society IEEE*.

Pichler, R (2010a). *Agile Product Management with Scrum: Creating Products That Customers Love*. Addison-Wesley Professional.

Pichler, R. (2010b). *The Definition of Ready*. Geraadpleegd op 3 oktober 2013, van <http://www.romanpichler.com/blog/product-backlog/the-definition-of-ready/>.

Pichler, R. (2012). *The product backlog grooming steps*. Geraadpleegd op 3 oktober 2013, van <http://www.romanpichler.com/blog/product-backlog/the-product-backlog-grooming-steps/>.

Poppendieck, M. & Poppendieck, T. (2003). *Lean Software Development: An Agile Toolkit*. Addison-Wesley.

Saunders, M. e.a. (2011). *Methoden en technieken van onderzoek*. Amsterdam: Pearson Education Benelux.

Schwaber, K. & Sutherland, J (2011). *The Definitive Guide to Scrum: The Rules of the Game*. Geraadpleegd op 6 september 2013, van

[https://www.scrum.org/portals/0/documents/scrum%20guides/scrum\\_guide.pdf](https://www.scrum.org/portals/0/documents/scrum%20guides/scrum_guide.pdf).

Sommerville, I. (2011) *Software Engineering*. Boston: Pearson.

Sutherland, J. & Schwaber, K. (2007). *The Scrum Papers: Nuts, Bolts, and Origins of an Agile Framework*.

Geraadpleegd op 6 september 2013, van

[http://scruminc.com/tl\\_files/scrum\\_inc/documents/ScrumPapers.pdf](http://scruminc.com/tl_files/scrum_inc/documents/ScrumPapers.pdf).

Verschuren, P. & Dorenwaard, H. (2007). *Het ontwerpen van een onderzoek*. Den Haag: Lemma.

West, D. (2011). *Water-Scrum-Fall Is The Reality Of Agile For Most Organizations Today*. Forrester

Research, Inc.

### **Tabellen en Figuren**

scaledagileframework.com

leadinganswers.typepad.com

scruminc.com

# Bijlagen

## Bijlage 1: Topics en eventuele vragen

### Topic 1: Algemeen

- Wat doet de organisatie? Core business?
- Hoe ziet de organisatie er uit? Structuur?
- Hoe groot is de ICT-afdeling?
- Wat is jouw functie in het bedrijf?
  - Rol
  - Taken
  - Verantwoordelijkheden
- Bij hoeveel projecten ben je op dit moment betrokken?
  - Wat vind je van deze hoeveelheid?
- Hoe vaak worden hier projecten uitgevoerd?
  - 0-5    5-10    10> (per maand?)
- Hoe ziet het project er uit) Kenmerken van de projecten?
  - (Waar gaat het over?)
  - (Gemiddelde) doorlooptijd: <3 mnd    3-6mnd    >6mnd
  - (Gemiddelde) grootte van team: <3 mensen    3-9 mensen    >meerdere teams
  - Over hoeveel panden, afdelingen, locaties, steden, landen loopt dit project?  
Zijn er meerdere timezones in het project? Is er sprake van virtuele teams? 1 afdeling, meerdere afdelingen, meerdere locaties
  - Wat is (meestal) de ervaring van het project team met de gebruikte technologieën en werkwijze?
- Wordt er altijd volgens Scrum gewerkt?
- Hoe werd er gewerkt voordat Scrum kwam?

### Topic: 2 Hoe komt de Product Backlog tot stand

- Hoe wordt een nieuw project gestart beschrijving van start tot aan eerste sprint?
  - Gebeurt dit door middel van een voortraject/ een gestandaardiseerd proces?
  - Gebeurt dit nog veel volgens de watervalmethode? Is er een sprint 0?
  - Wat gebeurt er precies op het gebied van analyse en ontwerp?
    - Welke onderdelen/stappen worden er doorlopen?
- En wat verder? analyse documenten bepalen etc? Hoe doe je dat allemaal?
  - Welke rollen en verantwoordelijkheden zijn er?
  - Wie neemt de beslissingen?
  - Wat leveren zij dan op? Documenten? User stories
  - Hoe ziet een user story er dan uit? Hoe zien die documenten er uit?
  - Hoe lang duurt dit proces? Ligt er vast wanneer dit proces klaar is? Wanneer je begint met een sprint? Is er een checklist?
  - Hoe bepaal je dit allemaal? Hoe bepaal je wat er moet gebeuren in het voortraject op het gebied van analyse en ontwerp? Zie vraag 8
- Is dit een goede aanpak of kan dit anders/beter? En zo ja, hoe dan?

### Topic: 3 Wat is de rol van de informatieanalist

- Welk proces doorloopt de analist?
- Hoeveel lagen van analyse zijn er?
- Wat is de input/output van de analist?
- Welke plek heeft de analist, onderdeel van het team, naast Product Owner, als Product Owner etc.?
- Hoe verhoudt deze rol zich met die van de Product Owner?

- Hoe verhoudt deze rol zich tot de Functioneel Ontwerper?
- Hoe verhoudt deze rol zich tot het team?
- Welke technieken gebruikt hij? Hoe maakt hij een keuze om bijvoorbeeld een ERD te maken of niet? Komt er een ontwerper kijken bij het uitbreiden en specificeren van de User Stories of is dit de taak van de analist?
- Wanneer speelt de analist een rol? In het voortraject? Tijdens de sprint? Tijdens Backlog Refinement? Altijd?
- Is dit een goede invulling voor de analyse rol binnen Scrum? Hoe moet die eventueel anders?

#### Topic 4: Hoe gebeurt ready maken

- Is dit onderdeel van het voortraject?
- Wie is hiervoor verantwoordelijk?
- Is dit een apart proces?
- Is er duidelijke welke User Stories ready zijn, welke niet ready zijn en aan welke wordt gewerkt?
- Hoeveel worden er ready gemaakt? Hoe wordt dit bepaald?
- Waar bestaat ready maken uit? Ontwerpen?
- Begint een project wel eens met uitsluitend user stories (dus alleen tekst)?
- Blijkt het wel eens dat een User Story (inclusief uitwerking) onvoldoende is tijdens een sprint? Wat doe je dan?
- Wordt er op voorhand gecheckt in bijvoorbeeld de Definition of Ready en/of Ready items of een team voldoende heeft om met de story aan de slag te gaan?
- Wat gebeurt er tijdens het proces op het gebied van analyse en ontwerp?
- Wie is hier verantwoordelijk voor en wie doet dit uiteindelijk?
- Wordt backlog grooming gebruikt/ vorm van requirement management?
- Zo ja: Hoe gaat dit in zijn werk? Wie is hier bij betrokken en wat gebeurt er in zo'n sessie?
- Hoe zien User stories en de DoR/ DOD er uit?
- Welke richtlijnen zijn er voor het schrijven van een User story?
- Hoe worden afhankelijkheden duidelijk gemaakt?
- Nog iets anders dat gebeurt met de User Stories? Traceerbaarheid?
- Wie is hier voor verantwoordelijk en wanneer gebeurt dit?
- Wanneer wordt dit bepaald (de definition of Ready opgesteld?) en door wie?
- Waaruit bestaat deze? Voorbeeld?
- Zijn documenten onderdeel van de Definition of Ready?
- Gaat de Definition of Ready (de input voor een sprint) alleen uit van tekst of kan onderdeel van de Definition of Ready ook bijvoorbeeld het maken van een wireframe en/ERD etc zijn? (ontwerp deel)
- Hoe zit het met de Definition of Done?

#### Topic 5: Slot

- Is de aanpak op het gebied van analyse en ontwerp voor iedere opdracht gelijk? Zowel in het voortraject als tijdens een sprint?
- Wat zijn knelpunten en succesfactoren?

## Bijlage 2: Samenvatting resultaten per case

Case	Het komen tot de (Initiële) Product Backlog
<b>Case1</b>	De Product Backlog is verkregen door het proces te beschrijven en te doorlopen. Aan de hand daarvan zijn User Stories benoemd samen met Stakeholders. Het doorlopen van het proces gebeurde aan de hand van een klikbaar prototype dat in een voortraject is gemaakt.
<b>Case2</b>	De Product Backlog is gevuld door op hoog niveau scenario's te beschrijven. Deze scenario's zijn Epic's. Aan de hand daarvan zijn User Stories geformuleerd per Epic. Het formuleren van de User Stories is dus niet op voorhand gedaan maar gebeurde voorafgaand aan een sprint op basis van (Business)analyse documenten en de beschreven Epics. Op deze manier komt de Product Backlog gedurende het project tot stand. Het vooraf neerzetten van de Product Backlog op User Story niveau is te lastig gebleken voor het project van wege de complexiteit.
<b>Case 3</b>	Vanuit Business Analyse en informatieanalyse (documenten) en eerder opgestelde Use Cases is het proces beschreven en inzichtelijk gemaakt op een procesplaat. Door het proces te doorlopen met Business Analisten en informatieanalisten zijn User Stories benoemd en is de Initiële Product Backlog gevuld.
<b>Case 4</b>	Vanuit Business Analyse en informatieanalyse (documenten) en eerder opgestelde Use Cases zijn Epic's beschreven. Dit heeft geleid tot een High Level Product Backlog. Vervolgens zijn de Epic's aan een team toegeschreven en werden er voorafgaand aan de eerste sprint User Stories geformuleerd. Dit gebeurde door de Product Owner en de Functioneel Ontwerper.
<b>Case 5</b>	Door processen te beschrijven, de meest essentiële functionaliteiten te benoemen en hiervoor User Stories te formuleren is de Initiële Product Backlog opgesteld.
<b>Case 6</b>	High level requirements vanuit de business werden door het team omgebouwd naar User Stories. Dit ging vrij Adhoc. Een nieuwe ontwikkeling is om met stakeholders het proces High Level te doorlopen en aan de hand daarvan User Stories te benoemen.
<b>Case 7</b>	R&D-team heeft zelf de Product Backlog gevuld door vanuit stakeholders te kijken naar welke functionaliteiten er moesten komen. De Product Backlog voor het formulierenteam is verkregen door het omschrijven van het formulier dat gemaakt moest worden. Het ging hier om een erg concreet project.
<b>Case 8</b>	Er waren verschillende Product Backlogs. De initiële Product Backlog werd gevormd door processen te benoemen en aan de hand van deze processen Epics te beschrijven in een pre-ready team op hoog niveau. Deze processen zijn inzichtelijk gemaakt op een High Level procesplaat. De Epic's werden verder onderzocht door het Ready-team. Het Ready-team voerde analyse uit en beschreef processen gedetailleerder om hier User Stories uit te kunnen formuleren. Op deze manier werd de Product Backlog op User Story niveau verkregen die vervolgens het done-team in gingen.



Case	Het ready maken
<b>Case1</b>	<p>Het ready maken was een apart proces dat parallel liep aan de huidige sprint met als doel huidige en nieuwe User Stories verduidelijken, verder uitwerken en ontwerpen maken voor de komende sprints. Daarnaast hield het in dat items die niet ready waren dit als nog werden tijdens de sprint. De detaillering hiervan varieerde. Dit was afhankelijk van de complexiteit van de User Story. Wat de sprint in ging kon alleen een User Story zijn of het kon een meer Use Case-achtig formaat aannemen. De Use Case werd in dat geval een los attachment naast de User Story. Er werd gecheckt bij de gebruikers en bouwers of dit volstond. De wens voor specificatie varieerde sterk in het team. Uiteindelijk werd er wel altijd een Functioneel Ontwerp opgesteld omdat dit een documentatie-eis was maar dit was onderdeel van de Definition of Done. Het was lastig om het ready maken bij te houden. Dit kwam vooral door een opstapeling van rollen en verantwoordelijkheid bij één persoon.</p>
<b>Case2</b>	<p>Ready maken gebeurde eigenlijk voor twee momenten ten eerste moest de Epic ready zijn vanuit de business analyse richting het voorbereidingsteam. Ten tweede maakte de informatieanalist uit het voorbereidingsteam de User Story ready door deze zo te beschrijven dat het voor iedereen duidelijk was. De input hiervoor waren de analyse documenten die opgeleverd werden vanuit Business Analisten. Daarnaast was er het moment dat de Functioneel Ontwerper de User Story had uitgewerkt zodat de developer aan de slag kon. Wat dan de behoefte was varieerde sterk. In de teams zaten zowel mensen die een volledige specificatie wilden ontvangen als mensen die het liefst met alleen een User Story aan de gang gingen. Uiteindelijk werd alles in een Use Case beschreven. Dit was onderdeel van de Definition of Done. Wat betreft het ready maken van User Stories werd het lastig gevonden om voldoende werkvoorraad te creëren.</p>
<b>Case 3</b>	<p>Ready maken gebeurde door de informatieanalist in het overkoepelend team. Dit hield in dat de User Story ready gemaakt werd door deze zo te beschrijven dat het voor iedereen duidelijk was. Dit gebeurde door onder andere een beschrijving van de regels en verwijzingen naar documenten waar dingen stonden. Tijdens de sprint was er de Functioneel Ontwerper die de User Story had uitgewerkt zodat de developer aan de slag kon. Voor dit deel bestond er een vast formaat namelijk een Use Case. De developers hadden deze Use Case in de meeste gevallen nodig om aan de slag te kunnen. Daarnaast was de Use Case een documentatie eis. De Functioneel ontwerpers liepen redelijk gelijk met bouw en test, wat een lage werkvoorraad opleverde.</p>
<b>Case 4</b>	<p>Het ready maken werd gedaan door een Functioneel Ontwerper in de sprint. Voor de iets complexere User Stories werd er geprobeerd om iets voor te lopen. Voor het ready maken werd een Use Case formaat gebruikt. Het verwerken van de User Story in de Use Case was onderdeel van de Definition of Ready.</p>
<b>Case 5</b>	<p>Het ready maken bestond uit een beschrijving van drie tot vier pagina's waarin werd beschreven wat er moest gebeuren. Dit was een soort low-level Use Case. Er was iemand verantwoordelijk om te zorgen dat het team de juiste informatie ontving vanuit het ready-team. Wanneer er te weinig ready was, ging het ervaren ontwikkelteam zelf er achteraan waardoor een deel van het ready maken lange tijd ook door het ontwikkelteam tijdens de sprint werd gedaan. Dit was een bewuste keuze om op deze manier wel voortgang te</p>

---

	kunnen boeken met het project.
<b>Case 6</b>	Het ready maken bestond uit het maken van een Use Case. Er werd aangegeven dat dit nog vrij Waterval gebeurde en dat dit lastig was wanneer er tijdens de sprints iets fundamenteels bleek te ontbreken. Het maken van een Use Case was onderdeel van de Definition of Ready.
<b>Case 7</b>	Het ready maken bestaat uit het maken van een Use Case. Dit gebeurde door eerder gemaakte Use Cases aan te passen.
<b>Case 8</b>	Ready maken gebeurde eigenlijk voor twee momenten ten eerste door het pre-ready team dat Epic's beschrijft. Ten tweede vanuit het ready-team richting het done-team. Wat er moest gebeuren om een User Story ready te maken varieerde. Er werd gekeken naar wat de behoefte was vanuit het team. Dit kon alleen een User Story zijn maar dit kon ook een uitgebreidere beschrijving zijn. Hiervoor werden elementen uit Use Cases gebruikt wanneer verdere specificatie nodig is.

---

Case	De rol van de informatieanalist
<b>Case1</b>	<p><b>Plaats:</b> Onderdeel van het team en bezig met het voorbereiden van de komende sprint.</p> <p><b>Rol:</b> Ready maken van User Stories en het maken van de ontwerpen.</p>
<b>Case2</b>	<p><b>Plaats:</b> Onderdeel van een voorbereidingsteam dat tussen de Business Analisten en het realisatieteam in zit. De Business Analisten in de organisatie houden zich bezig met het opleveren van documenten en zijn niet betrokken bij de realisatie.</p> <p><b>Rol:</b> User Stories beschrijven en deze ready maken voor de komende sprints.</p>
<b>Case3</b>	<p><b>Plaats:</b> Overkoepelend team met informatieanalisten en Business Analisten buiten de realisatieteams.</p> <p><b>Rol:</b> User Stories beschrijven en ready maken.</p>
<b>Case4</b>	<p><b>Plaats:</b> Los van het realisatieteam</p> <p><b>Rol:</b> Epic's beschrijven aan de hand van informatieanalyse. Overzicht bewaken tijdens de realisatie.</p>
<b>Case5</b>	<p><b>Plaats:</b> In een apart ready-team.</p> <p><b>Rol:</b> Beschrijven van User Stories en maken van een low-level Use Case als input voor de komende sprint.</p>
<b>Case 6</b>	<p><b>Plaats:</b> Onderdeel van het team maar loopt wel voor op de sprint.</p> <p><b>Rol:</b> User stories en Use Cases maken.</p>
<b>Case 7</b>	<p><b>Plaats:</b> In een apart ready-team</p> <p><b>Rol:</b> Functioneel ontwerpen aanpassen (Use Cases) en opleveren</p>
<b>Case 8</b>	<p><b>Plaats:</b> In een ready-team los van het done-team</p> <p><b>Rol:</b> User stories verduidelijken en ontwerpen maken voor de komende sprints.</p>

## Bijlage 3:Definition of Ready

### Definition of Ready 1

User Story:

- User Story is gereviewd (zie onder)
- User Story is ingeschat

Product Backlog:

- Backlog in Jira is geprioriteerd
- Backlog in Jira is ingeschat

### Review

- Duidelijk in welke service en welke deployable unit
- Bij uitsplitsing duidelijk aan welke PB-story deze hangt
- Inhoudelijk afgestemd met P.O.
- Verwerkt in functioneel ontwerp (use case en eventuele andere functioneel ontwerpen):
  - Input, outputparameters
  - Schermbeschrijving
  - Logisch datamodel
  - Procesflow

### Definition of Ready 2

- Er een technische omgeving beschikbaar is;
- Het datamodel is bijgewerkt aan de hand van de user story;
- Er een mapping document bij de user story bestaan met een mapping tussen de service/het scherm en het domeinmodel;
- Er een schermontwerp bij de user story bestaat;
- De user story is afgestemd met de business;
- De user story is geprioriteerd;
- De user story is ingeschat;
- Er een how to test is beschreven bij de user story (met evt verwijzingen naar documenten);
- In Jira bij de user story een link is geplaatst naar de bijbehorende documenten.

### Definition of Ready 3

- Wat gebouwd moet worden is helemaal ontworpen (beschreven in een Use Case)
- Dit moet gereviewd zijn door de Product Owner, een andere ontwerper, een tester en een developer.
- Ingeschat en geprioriteerd
- Test staat beschreven

### Definition of Ready 4

De User Story is helder, duidelijk en ondubbelzinnig gedefinieerd

Hierbij kan de User Story getoetst worden aan de INVEST criteria:

- Independent: de User Story is onafhankelijk van andere stories te realiseren  
Als dit is niet mogelijk is omdat de story afhankelijk is van andere stories in die sprint of zelfs stories uit dezelfde sprint binnen andere teams dan moet afgesproken worden hoe de afhankelijke stories op done komen.
- Negotiable: het team bespreekt met de product owner wat wel en niet binnen de User Story valt

- Value: de User Story vertegenwoordigd business value (anders is het geen story maar een taak)
- Estimable: de omvang van de User Story is te schatten
- Small: de User Story past makkelijk binnen een sprint (bijvoorbeeld maximaal 4 dagen of 5 punten)
- Testable: de User Story is (onafhankelijk) te testen
- De functionaliteit van de User Story is duidelijk beschreven.  
Het gaat er om dat het realisatieproces niet verstoord wordt door discussies; afstemming is natuurlijk prima maar een story die je gestart bent moet je ook af kunnen maken zonder op anderen te hoeven wachten.
- In welke vorm de functionaliteit beschreven is om voldoende duidelijk te zijn voor realisatie is afhankelijk van het onderwerp (regels, schermen, BackOffice processen) en kan per team verschillen.
- De User Story kent een duidelijk begin en eindpunt.  
Het is duidelijk wat de begin en de eind conditie zijn van de story; het is duidelijk wat de resultaten zijn en hoe deze vastgesteld (getest) kunnen worden.
- De acceptatiecriteria zijn duidelijk en SMART.
- Vooraf is bekend wanneer de User Story 'af' is: dit kan gedefinieerd zijn tijdens de analyse of afgesproken worden met de product owner. Bijvoorbeeld: "Het inlezen van het MT940 bestand is klaar als 98,5% van de transactiecodes van testbestand 'X' succesvol geparsed worden"

### Definition of Ready 5

#### Definition of Pre-Ready:

- High Level User story (ready & explainable in 5 min)
- Acceptance Criteria
- High Level Solution in set
- Systems (which systems are involved)
- Information needs (on Business Object Model level)
- Process Flow and Description
- Key Requirements mapping is done

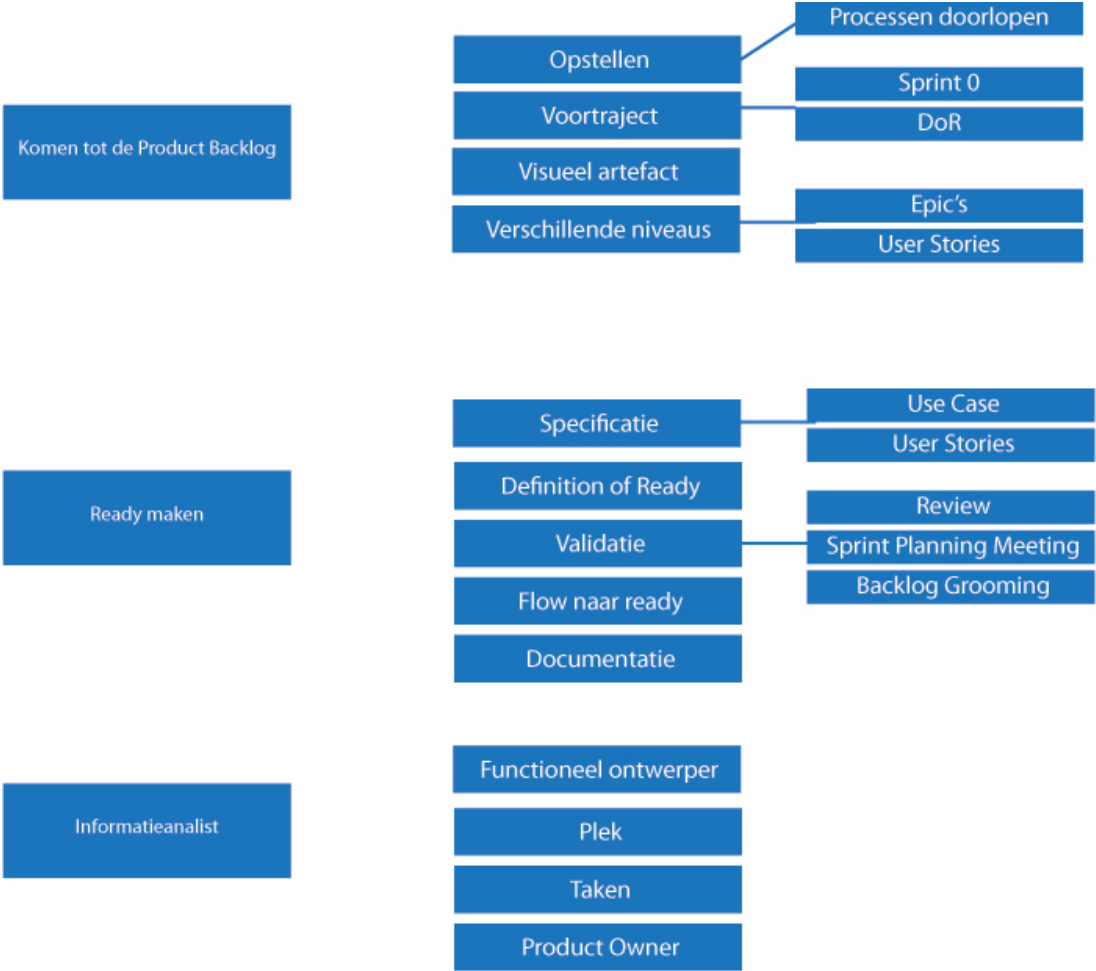
#### Definition of Ready:

- User Story is explainable in 5 minutes
- All parts of the User Story template are written or explained why they are empty
- Interfaces to surrounding systems are clear and defined in stubs Granularity is approx. 2-3 days of development work (for 1 person)

## Bijlage 4: Categorieën



# Bijlage 5: Uiteindelijk categorieën per thema



## Bijlage 6: Richtlijnen analyse en ontwerp in een Scrumproject

In deze bijlage zijn de uiteindelijke richtlijnen te vinden die onderdeel zijn van het uiteindelijke advies aan Quintor. De richtlijnen zijn opgesteld aan de hand van het referentiekader (paragraaf 2.2) en aan de hand van het advies in relatie tot de bevindingen (paragraaf 4.2)

### Het komen tot de Product Backlog

#### Opstellen

- Het opstellen van de Product Backlog dient te gebeuren aan de hand van de meest essentiële functionaliteit.
- Het creëren van een bepaalde context tijdens het opstellen van de Product Backlog is essentieel.
- Om te komen tot de Product Backlog dienen de volgende stappen te worden doorlopen voorafgaand aan de eerste sprint:
  1. Visie bepalen
  2. Actoren bepalen
  3. Functies per Actor in kaart brengen
  4. Functies prioriteren
  5. Belangrijke processen beschrijven zodat tijdens het vormen van User Stories er functionaliteiten geformuleerd kunnen worden
  6. Interfaces bepalen zodat duidelijk is met welke systemen het project te maken heeft
  7. User Stories formuleren
  8. Opstellen Product Backlog, prioriteren en inschatten

#### Voortraject

- Er moet een sprint 0/voortraject plaatsvinden waarin naast het opstellen van de Product Backlog de Definition of Ready en Definition of Done wordt bepaald en User Stories voor de eerste 1 à 2 sprints ready gemaakt worden.

#### Visueel Artefact

- Er kan een visueel artefact dat (high level) de processen van het te bouwen systeem weergeeft worden opgesteld in het voortraject/sprint 0. Op deze manier kan de context worden verduidelijkt en kunnen User Stories worden benoemd. Het visueel artefact creëert tevens een gemeenschappelijk taal waarmee voortgang kan worden getoond en vertrouwen van stakeholders kan worden gewonnen. Het hebben van een visueel artefact nodigt uit om foute te benoemen waardoor er constant feedback wordt geleverd. Door het geen officieel document te maken geeft het veel vrijheid.
- Het visueel artefact kan up-to-date worden gehouden aan de hand van nieuwe inzichten zodat het een middel blijft gedurende het project.

#### Verschillende niveaus

- Wanneer er bij grote projecten en/of organisaties sprake is van het beschrijven van Epic's kan er worden overwogen om User Stories en teams op verschillende niveaus te beschouwen zoals in



het Scaled Agile Framework wordt gedaan. Dit heeft als doel om ervoor te zorgen dat er geen te grote scheiding ontstaat tussen het beschrijven van Epic's op hoog niveau en de realisatie volgens Scrum.

## Het ready maken van User Stories

### Flow naar ready

- Het ready maken van User Stories dient als een apart proces te worden beschouwd en dient te worden gezien als een Flow naar Ready. Op deze manier kan continu werkvoorraad worden gecreëerd voor de komende sprints.
- De User Stories dienen in verschillende categorieën te worden ingedeeld om de fase waarin de User Story verkeert inzichtelijk te maken: 'Open for design' 'In design' 'Testable' en 'Ready'. Dit maakt het proces van ready maken voor iedereen duidelijk en open voor feedback.
- Wanneer User Stories onvoldoende ready blijken dient hierop geacteerd te worden, hetgeen kan leiden tot veranderingen in het proces waarin de Flow naar ready wordt uitgevoerd.
- Dit proces dient te worden uitgevoerd voor de eerste sprint en tijdens sprints. Tijdens de sprint is dit een parallel proces aan de huidige sprint met als doel iedere keer weer voldoende stories ready te hebben voor de komende sprint.
- Het is ook tijdens sprints belangrijk om 1 à 2 sprints vooruit te werken en uitsluitend grote lijnen te bepalen en niet alles tot in detail uit te werken.

### Validatie

- Een User Story dient voldoende informatie te bevatten zodat een ontwikkelaar aan de slag kan. Om te bepalen wat dit is, moet dit gecheckt worden aan de hand van de Definition of Ready. . Hoewel validatie noodzakelijk is geeft het niet altijd zekerheid omdat vaak, ondanks een validatie, User Stories niet ready blijken. Het is daarom belangrijk om dit onderdeel niet te zwaar te maken en tijd over te houden om tijdens de sprint problemen die ontstaan op te lossen.

### Definition of Ready

- De Definition of Ready moet worden opgesteld bij de start van het project en dient te worden nageleefd zodat hier een algemeen begrip over heerst en alleen ready User Stories de sprint in gaan. Ready betekent User Stories conform deze definitie opleveren.
- Gedurende het project moet er worden gekeken of de Definition of Ready wel realistisch is en moet deze zo nodig worden bijgesteld. Op deze manier wordt Just-in-time en Just-Enough-analyse-en-ontwerp mogelijk.
- De Definition of Ready bevat in essentie vaak in ieder geval de volgende onderdelen:
  - Geprioriteerd
  - Inschatbaar
  - Gereviewd
  - How to test is opgesteld

- Klein genoeg/opgesplitst
- Voor iedereen duidelijk

## Specificatie

- Om User Stories van voldoende informatie te voorzien kan gebruik worden gemaakt van hulpmiddelen zoals het opstellen van Wireframes en Flowcharts.
- Het uitbreiden van de User Story kan goed gebeuren aan de hand van elementen vanuit de Use Case. Hierbij is het belangrijk om te allen tijde naar de noodzaak te blijven kijken.
- User Stories moeten 'Just-Enough' worden gespecificeerd en 'Just-in-Time' klaar zijn aan het begin van een sprint. Het opstellen en hanteren van de Definition of Ready draagt hier aan bij.
- Een invulling van de specificatie moet niet gezocht worden in traditionele manieren van werken (de Watervalmethode) maar dient zoveel mogelijk op een Agile/Scrum manier te worden ingevuld om zo de hoeveelheid waste te beperken.
- Factoren als kennisniveau/informatiebehoefte van het team, kennis in de organisatie kennis die nodig is bij het realiseren van de IT-ondersteuning, afhankelijkheden van andere systemen, een bepaalde cultuur, geen goede Product Owner, documentatie-eisen en gescheiden locaties zorgen ervoor dat het ready maken omvangrijker wordt. Hier dient men op te acteren tijdens de Flow naar ready en bij aanvang van het project bij stil te staan.

## Documentatie

- Binnen Scrum moet, wanneer er een invulling wordt gezocht voor analyse en ontwerp, de focus niet liggen op het opleveren van documenten, maar op het hoe het systeem zich uiteindelijk moet gedragen.
- In een voortraject is het belangrijk de documentatie-eis te beperken tot een minimum en hierbij
- Er dient te worden overwogen om documentatie op te nemen als onderdeel van de Definition of Done en niet van de Definition of Ready. Dit is verstandig omdat anders het gevaar bestaat dat er te vroeg, te veel, wordt gedaan. Vervolgens moet het dan later aangepast worden, omdat de documentatie dan (deels) niet meer aansluit bij hetgeen dat gebouwd is.

## De rol van de informatieanalist

- De rol van een informatieanalist mag binnen Scrum niet worden vergeten. De Product Owner dient ondersteund te worden door een informatieanalist wanneer het project hier om vraagt.
- Het is belangrijk om te waarborgen dat de informatieanalist puur ter ondersteuning van de Product Owner dient en niet ter vervanging. De Product Owner heeft specifieke belangen, kennis en mandaat en dient daarom de verantwoordelijkheid te hebben en te nemen.
- Het ready maken van items is de verantwoordelijkheid van de informatieanalist en de Product Owner.
- Met betrekking tot het ready maken is het niet zinvol tussen informatieanalisten en functioneel ontwerpers een scheiding aan te brengen

- De informatieanalist moet de persoon zijn die de discussie tussen het team en de Product Owner faciliteert. De informatieanalist is hiervoor de juiste persoon omdat deze beschikt systeemkennis en in staat is om systeemeisen boven water te krijgen en te specificeren.
- De informatieanalist moet in staat zijn om 'Just-Enough' te specificeren en 'Just-in-Time' klaar te zijn.
- De informatieanalist is ook tijdens het project betrokken bij het managen van de Product Backlog in de vorm van Backlog Refinement.
- De plek de informatieanalist is op de rand. Tussen het team en de Product Owner in. Dat betekent dat deze zowel meewerkt in de sprint als in de Flow naar Ready. Bij kleine projecten kan de plaats van de informatieanalist in het team zijn.
- De informatieanalist is te allen tijde beschikbaar voor het team zodat deze betrokken blijft voor het beantwoorden van vragen en het behalen van het sprintdoel.