Builders & Performers

Plan van Aanpak

Afstudeeropdracht van Ruben van Stralen

Afstudeerder:	Ruben van Stralen
Studentnummer:	12072400
School:	De Haagse Hogeschool te Zoetermeer
Opleiding:	Business IT & Management
Opdrachtgever:	Ronny Ramdhan
Begeleider:	Henk Hoeksema
Organisatie:	Builders & Performers
Periode:	08-Februari 2016 tot 3-Juni 2016

Inhoudsopgave

1. Inleiding

Voor het afstudeerproject bij Builders & Performers met opdrachtgever Ronny Randham, is een Plan van Aanpak geschreven als leidraad. Het afstudeerproject is begonnen op 8 februari en zal rond begin juni eindigen. Het plan van aanpak bestaat uit de volgende onderdelen:

Hoofdstuk 2, Achtergronden:	Hierin wordt verteld wat de aanleiding van het project is, wordt de probleemstelling gedefinieerd met de bijhorende hoofdvraag en de doelstellingen.
Hoofdstuk 3, Projectaanpak:	Voor het project is gekozen om met de methode DMAIC te werken. Per fase wordt er beschreven wat er globaal in wordt gerealiseerd. Daarnaast wordt ook beschreven hoe het onderzoek wordt uitgevoerd en welke methodes daarbij zijn gekozen.
Hoofdstuk 4, Planning, projectactiviteiten en resultaten:	De planning voor de komende periode wordt hierin beschreven met per fase welke activiteiten er worden uitgevoerd en welke resultaten het opgeleverd.
Hoofdstuk 5, Kwaliteit & Scope:	Om te zorgen dat het de resultaten kwaliteit opleveren, wordt er per onderdeel beschreven hoe dit wordt gewaarborgd. Ook is er een scope beschreven waarin staat vermeld wat wel en niet als eindresultaat wordt geleverd.
Hoofstuk 6, Stakeholders:	Met behulp van het boek " De kunst van veranderen" is er stakeholders analyse gemaakt. De analyse geeft inzicht welke stakeholders betrokken moeten worden in het project.
Hoofdstuk 7, Randvoorwaarden en Risico's:	In dit hoofdstuk wordt aangegeven wat de randvoorwaarden van het project zijn en welke risico's er kunnen optreden. Bij de risico's is ook aangeven welke maatregelen er genomen kunnen worden.
Hoofdstuk 8, Communicatie:	Om te zorgen dat belanghebbende van het project niet voor verassingen komen te staan, is er beschreven hoe er gezamenlijk wordt gecommuniceerd.

2. Achtergrond

2.1 Aanleiding

Builders & Performers is een organisatie die zorgt voor de ontwikkeling en implementatie van softwareoplossingen bij de klant. Op basis van een eigen standaardpakket genaamd EASE met de mogelijkheid tot klantspecifieke aanpassingen. Om te zorgen dat de klant optimaal geholpen wordt, heeft Builders & Performers een team samengesteld met een BPM specialist en ontwikkelaars. De BPM specialist heeft de rol om bij de klant de betrokken processen te inventariseren en de business requirements in kaart te brengen. De requirements die niet met het standaardpakket worden opgelost, worden vervolgens als specificaties opgeleverd aan de ontwikkelaars. De ontwikkelaars realiseren vervolgens een gewenste softwareoplossing aan de klant. Als laatste neemt de BPM specialist een acceptatietest door met de klant om tot een goedkeuring te komen.

Een nauwe samenwerking tussen de BPM specialist en ontwikkelaars speelt zodoende een cruciale rol bij een succesvolle oplevering. Ronny Randham, de opdrachtgever, heeft daarom gevraagd om een onderzoek te uit te voeren welke methodes/hulpmiddelen kunnen bijdragen aan een efficiëntere samenwerking tussen een BPM specialist een de ontwikkelaars. Dit geldt zowel voor projecten bij nieuwe klanten als voor gevraagde wijzigingen bij bestaande klanten.

Het bedrijf Builders & Performers is op zoek naar een succesvolle formule van project- en softwareontwikkelingsmethodes, rollendefinities en een efficiënte kennisdeling tussen de WAT-vraag en de HOE-vraag.

2.2 Probleemstelling

De samenwerking tussen een BPM-specialist en een ontwikkelaar gaat momenteel niet efficiënt genoeg binnen Builders & Performers. Dit heeft tot gevolg dat de kwaliteit van de software niet voldoende is, kosten te hoog zijn, de oplevering niet optimaal verloopt en de betrokkenen zich niet in het proces thuis voelen. Op de manier hoe er nu wordt gewerkt, is de organisatie niet in staat verder te groeien. De organisatie krijgt binnenkort een aantal grote projecten toegewezen, verandering in de organisatie is daarom noodzakelijk om de projecten succesvol te laten verlopen.

2.3 Hoofdvraag

Welke oplossingen zijn er nodig om een efficiëntere samenwerking te bereiken tussen de BPM specialist en de ontwikkelaars?

2.4 De doelstelling/opdracht

Het doel van de opdracht is om met behulp van methodes en technieken het huidige software ontwikkelingsproces te verbeteren, waardoor binnen de organisatie efficiënter software wordt ontwikkeld. Het verbeterde software ontwikkelingsproces moet onder andere zorgen voor de volgende resultaten:

- Betere kwaliteit van de software vanuit het oogpunt van de klant
- Snellere oplevering bij de klant
- Betere communicatie tussen ontwikkelaars & BPM-ers/klanten
- Verhogen van het rendement
- Structuur binnen de de organisatie
- Ontwikkelaars en BPM specialisten zich beter thuis voelen in de methode/werkwijze

De opdracht is om het in kaart brengen van de huidige situatie, de problemen en oorzaken, de requirements van de betrokkenen te achterhalen, en de gewenste situatie in kaart brengen om tot een verbeterde softwareontwikkeling proces te komen en de gap tussen de BPM specialisten en de ontwikkelaars te verminderen. Hierbij formuleer ik een handboek voor B&P en een implementatieplan om het handboek in te voeren om de gap te dichten en de eerste Quick Wins te definiëren voor de organisatie. Na het verbetervoorstel aan Builders & Performers worden de eerste “quick wins” gerealiseerd uit het implementatieplan.

2.5 Deelvragen

Aan de hand van de hoofdvraag zijn er deelvragen gedefinieerd. De deelvragen moet gezamenlijk een antwoord geven op de hoofdvraag.

I. Waarom wilt de organisatie veranderen?

II. Wat is de huidige situatie van de organisatie in dit proces?

III. Wat is de gewenste situatie binnen de organisatie van dit proces?

IV. Wat is er nodig om de GAP te dichten tussen de huidige en gewenste situatie?

V. Wat zijn de definities van de software ontwikkelingsparadigma's?

VI. Welke software ontwikkelingsmethodes zijn er op de markt ?

VII. Wat zijn er de voor- en nadelen van verschillende software ontwikkelingsmethodes?

VIII. Welke andere technieken, methodes of hulpmiddelen zijn geschikt voor een efficiëntere samenwerkingsverband?

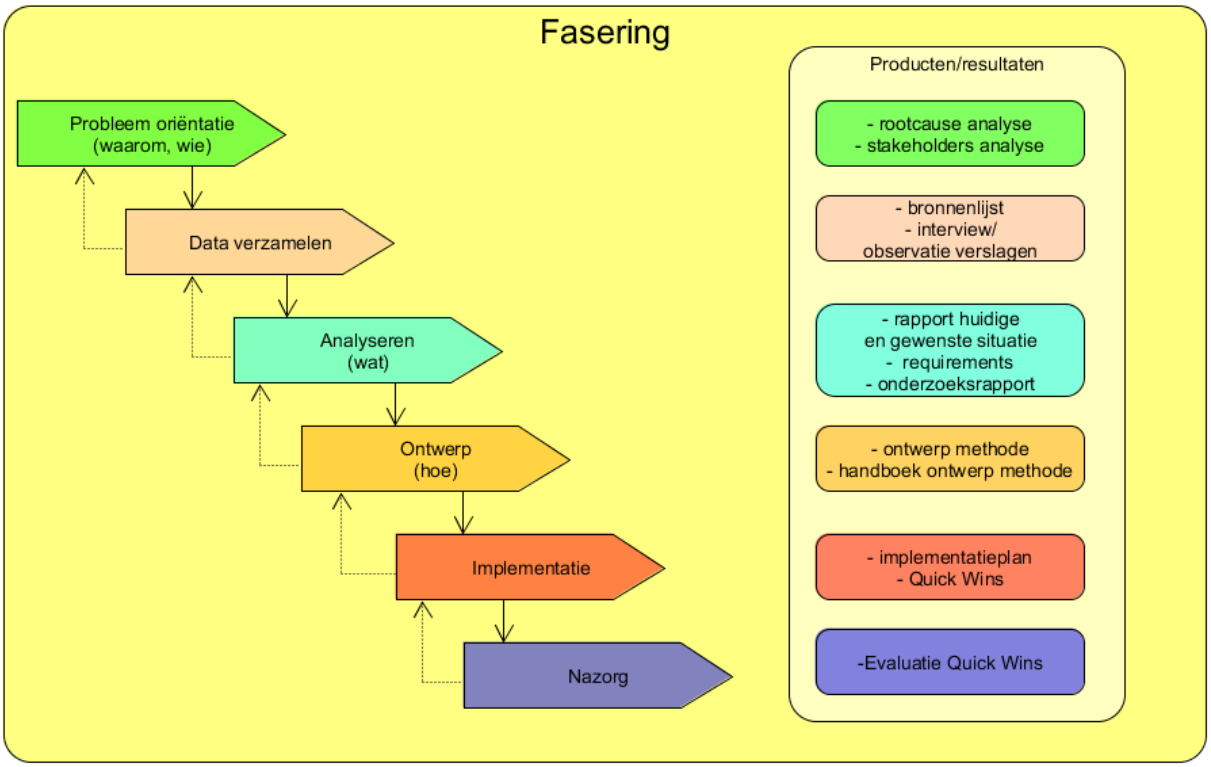
IX. Wat kan een software ontwikkelingsmethode of andere hulpmiddelen betekenen voor de samenwerking tussen een BPM specialist en ontwikkelaars?

3. Projectaanpak

Om te zorgen voor structuur en controle heb ik gebruik gemaakt van een fasering gebaseerd op het DMAIC model (van Vliet, 2014). Het DMIAC model wordt gebruikt om processen binnen een organisatie te verbeteren. Dit is ook het doel wat ik met deze opdracht wil bereiken. Met het DMAIC model wordt eerst het probleem gedefinieerd, vervolgens worden er gegevens verzameld. Daarna worden de verschillende oorzaak-relaties bekeken, wordt een creatieve oplossing bedacht en in de laatste fase wordt de oplossing geïmplementeerd. De laatste fase wordt afgerond met de evaluatie hiervan. De fasering van het DMAIC wordt deels gevolgd in dit project, bepaalde activiteiten worden vervangen. Als voorbeeld wordt het huidige proces niet gemeten (in de Measure fase), de eisen en wensen van de gebruikers worden in kaart gebracht en er worden gegevens verzameld over de verschillende software ontwikkelingsmethodes. De gehanteerde fasering resulteert voor dit project duidelijk wat er qua activiteiten wordt uitgevoerd en wordt geleverd resultaten.

Daarnaast maak ik gebruik van een iteratieve denkwijze. Een iteratiefproces zorgt ervoor dat ik een voorgaande fase bij verandering in de wensen en/of feedback van de gebruikers kan verbeteren en vernieuwen, in tegelstelling tot de watervalmethode.

Het project bestaat uit vijf fases:



Fase 1: Probleemoriëntatie

In de eerste fase ga ik het werkelijke probleem, de oorzaken en de gevolgen definiëren. Dit is in grote lijnen beschreven in hoofdstuk 3.2. In deze fase wordt samen met de opdrachtgever van de organisatie geanalyseerd wat het daadwerkelijke probleem is. Hiervoor worden interviews gehouden om tot de rootcause van het probleem te komen. Bij de interviews wordt er gebruik gemaakt van de 5WHY's methode van Lean Six-Sigma. De 5WHY's moeten ervoor zorgen om tot de rootcause van het probleem te komen. Ook worden de vragen gesteld waarom er veranderd moet worden. Aan de hand van de probleemstelling worden onderzoeksvragen opgesteld. Ook wordt in kaart gebracht wie de relevante stakeholders zijn en worden de business requirements van de opdrachtgever gedefinieerd.

Fase 2: Data verzamelen

In deze fase worden de gegevens verzameld om een beeld te creëren van de huidige en gewenste situatie. De gegevens worden verzameld met behulp van interviews, interventies en observaties. Het doel van de interviews is om in kaart te brengen wat er speelt in de huidige situatie bij medewerkers en wat de wensen zijn voor verbetering. Als interventie wordt er gebruik gemaakt van de employee empathie map, bekent als customer empathie map, om dieper inzicht te krijgen van de medewerker in o.a. gedrag, belang en aspiraties (Bland, 2016). Verder wordt de kleurentest van Caluwé gebruikt om te achterhalen in welke kleur de organisatie denkt en doet.

Bij de observaties wordt er gekeken naar de gebruikte tooling, artifacts en werkwijze van zowel de ontwikkelaars als van de BPM-specialisten in de betrokken processen. De processen en werkwijzen worden in kaart gebracht die betrekking hebben op het probleem met behulp van een flow diagram.

Daarnaast wordt er een onderzoek uitgevoerd naar de verschillende softwareontwikkelingsmethodes en de bijhorende technieken die gangbaar zijn in de markt. Die aanvullende kennis wordt gebruikt om het ontwikkelproces en het BPM-traject te verbeteren.

Fase 3: Analyseren

In deze fase worden de verzamelde gegevens geanalyseerd. Het hoofddoel van deze fase is om de gap te definiëren tussen de huidige situatie en de gewenste situatie. Uit de interviewverslagen, interventies en observaties wordt de huidige situatie en gewenste situatie vastgelegd van de organisatie. Daaruit worden vervolgens requirements gedefinieerd om in kaart te brengen wat de eisen en wensen van organisatie en de medewerkers van B&P voor het vernieuwde ontwikkelingsproces. In theorie worden requirements volgens Swart en IIBA opgesteld om een verbetering in een proces of een nieuw proces te realiseren met behulp van een systeem (de Swart, 2010). Echter kunnen requirements ook worden gebruikt om te definiëren wat de doelen zijn van een nieuw project waar de organisatie naar streeft (business requirements) en wat de gebruiker verwacht van de wijziging betrekking of het vernieuwde proces (gebruikers requirements) (Wiegers, 2003).

Met de verzamelde gegevens uit het literatuuronderzoek en de enquête wordt in kaart gebracht wat de verschillende gangbare softwareontwikkelingsmethodes zijn, wat de voor- en nadelen hiervan zijn,

welke technieken erbij worden gebruikt en welke softwareontwikkeling methodes aansluiten op de manier hoe de organisatie werkt en op de gedefinieerde requirements.

Met de vastgelegde analyse wordt met de organisatie van Builders & Performers een workshop gehouden om gezamenlijk te kiezen voor een geschikte oplossing(en) en/of softwareontwikkeling methode waarmee het ontwikkelproces en BPM-traject wordt verbeterd. Hierbij wordt ook de focus gelegd op eenvoudige en realiseerbare oplossingen.

Fase 4: Ontwerp

In fase 4 worden de gekozen oplossing(en) ontworpen met behulp van documenten, templates, visualisaties of andere hulpmiddelen welke een relevante waarde hebben voor de organisatie.

Fase 5: Implementatie

Na het ontwerp wordt in fase 5 een implementatieplan geschreven wat, hoe en door wie de implementatie wordt uitgevoerd. Het implementatieplan is gericht op het mens-leer traject en beschrijft de activiteiten die moeten worden uitgevoerd om de gekozen softwareontwikkeling methode zorgvuldig in te passen in de organisatie (Schop, 2006). Uit het implementatieplan worden vervolgens haalbare quick wins binnen de stageperiode gedefinieerd en vervolgens geïmplementeerd.

Na de oplevering en afronding van het project en na de stageperiode dient er blijvend aandacht te zijn voor de verbetering van het geïmplementeerde proces. Belangrijk is dat er op gezette tijden de geëvalueerd wordt en verbeteringen daaruit vervolgens worden opgepakt en geïmplementeerd. Dit leidt tot blijvende verbetering. Als de ingevoerde methode een vaste plek heeft gekregen in de organisatie, kan de laatste fase gesloten worden.

Fase 6: Nazorg

In de laatste fase worden de uitgevoerde Quick wins met de betrokkenen geëvalueerd. Het doel van de evaluatie is om te achterhalen of het beoogde resultaat is gerealiseerd bij en met de betrokkenen. Dit wordt gedaan met behulp van een scorekaart gecombineerd met een interview. Met de informatie uit de evaluatieverslagen wordt eventueel het ontworpen_proces/producten vernieuwd.

3.1 De vierballen methode

Naast de aanpak van DMAIC methode wordt de vierballen methode gebruikt als leidraad van de verandering. De vierballen methode bevat de vier kernvragen (waarom, wat, wie en hoe) die nodig zijn om het veranderidee in goede wegen te laten verlopen. Met het project wordt er linksom veranderd, oftewel Planned Change. Hierin wordt de volgorde gehanteerd van *waarom* → *wat* → *wie* → *hoe*. De waarom en wat vraag zorgen voor de ontwikkeling van het veranderidee (de Witte & Jonker, 2014).

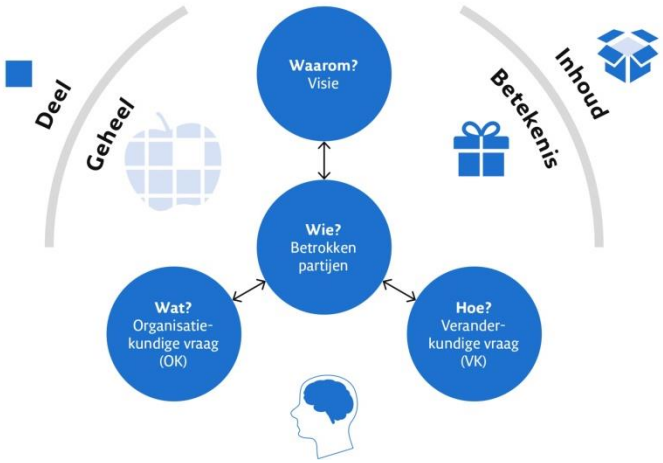
Bij de vragen van de vierballenmethode moeten o.a. deze vragen beantwoord worden:

Waarom: Waarom willen we veranderen? Wat moet het anders? Wat is de aanleiding?

Wat: Wat is de huidige en gewenste situatie(met de onderwerpen structuur, technologie, mens & cultuur)?

Hoe: Hoe geef je het veranderproces vorm? Wat moet er gerealiseerd worden om het veranderproces vorm te geven? Hoe gaan we de verandering benaderen?

Wie: Welke partijen worden in het veranderproces betrokken?



3.2 Methode onderzoek

Voor het project wordt er een kwalitatief onderzoek uitgevoerd. Het kwalitatieve onderzoek is een subjectieve methode om informatie te vergaren. Het doel van het kwalitatieve onderzoek is om de achterliggende motieven en meningen in kaart te brengen van de stakeholders. Ook is het doel om een verdiepend onderzoek te doen naar de verschillende softwareontwikkelingsmethodes. De resultaten van het onderzoek moeten richting geven op de oplossingen van het probleem.

Het onderzoek bestaat uit twee delen:

- Deskresearch
- Fieldresearch

Deskresearch

Bij de methode desksearch wordt er gebruikt gemaakt van bestaande gegevens. Hierbij wordt voornamelijk gezocht naar primaire en secundaire bronnen. Denk hierbij aan onderzoekrapporten of verslagen. De gegevens worden verzameld via:

- De databanken van Haagse Hogeschool en de Rotterdamse Hogeschool
- Google Scholar
- Gartner
- Google/overige zoekmachines
    - Via google/overige zoekmachines wordt als laatste gebruikt om het onderzoek mee uit te voeren i.v.m. de betrouwbaarheid en validiteit van de bronnen die erop te vinden zijn.

Om op gestructureerde wijze gegevens te verzamelen, wordt er gebruikt gemaakt van de Big-6 methode. De Big-6 methode bestaat uit de volgende stappen (The BIG 6, 2014):

1. Het probleem definiëren
2. De zoekstrategie kiezen
3. Informatie bronnen opsporen
4. Informatie verwerken
5. Informatie van meerdere bronnen samenbrengen
6. Evalueren

Om zoveel mogelijk relevante informatie te verzamelen is er gekozen om twee zoektechnieken te gebruiken voor het onderzoek. Dit zorgt niet alleen voor dat er meer gegevens worden verzameld, maar ook voor andere zoektermen en richtingen (van Veen & Westerkamp, 2012).

Sneeuwbalmethode: Hierbij begin je bij één relevante informatiebron. Daarin staat vaak bronnenlijst die het onderzoek ondersteunen. Deze bronnen kunnen vervolgens ook weer worden geraadpleegd om invulling te geven op het onderzoek. En via deze route zoek je ook weer verder naar de bronnenlijst.

Let wel op! Als er steeds verder gaat zoeken via de bron naar bron, wordt de bron steeds minder actueel.

Parelgroeien: Hoewel hieronder al een redelijke zoeklijst is opgesteld qua zoektermen, ben ik nog niet bekend met alle termen. Daarom probeer ik via deze methode eerst op de relevante termen te zoeken. Uit deze gevonden bronnen kan vervolgens worden gekeken naar andere relevante termen.

Hieronder is een lijst opgesteld met zoektermen die gebruikt kunnen worden tijdens het literatuuronderzoek (Software development process, 2014):

IT Proces improvement	Lean IT	Lean Software Development	Scrum
Agile	XP	CMMi	Human Centric Process Design
Prince 2	Proces herontwerp	Procesverbetering	Human centered Design
Software ontwikkelprocessen verbeteren	Lean CMMI	Rational Unified Proces	Projectmethodieken
Agile Unified Process	Watervalmethode	Babok	Software ontwikkelen
Iterative application Development	Request For Changes	V-model	SDN
Jackson System Development	Spiraalmodel	Feature Driven Development	IPMA
Test-Driven Development	Adaptive software Development	Dynamice systems Development Method	Rapid Application Development

System Development Methodology	MSP	PMW	PMC
PMBOK	Manifesto	Cobit 5	SAFE agile
Disciplined Agile			

Fieldresearch

Bij fieldresearch worden zelf gegevens verzameld met een eigen onderzoek. De informatie wordt verzameld via interviews, workshops, enquêtes en observaties. Het doel van het fieldresearch is om de probleemstelling vast te stellen en de huidige situatie en gewenste situatie vast te leggen (Fischer & Julsing, 2014).

De interviews kunnen op een gestructureerde, halfgestructureerde en ongestructureerde worden afgenomen. Voor het afnemen van de interviews bij dit project, wordt voornamelijk op een halfgestructureerde methode gewerkt. Hierin worden alleen de onderwerpen/hoofdvragen van het interview vastgesteld. Mijn eigen ervaring met het afnemen van interview is dat het altijd anders verloopt dan je verwacht. Daarom is het verstandig om van te voren de hoofdvragen/onderwerpen voor te bereiden, en tijdens het interview op verschillende momenten door te vragen of een discussie starten.

De functionaliteit om workshops te houden is om discussies te starten en gezamenlijk ideeën te creëren met verschillende stakeholders. Het levert vaak meer informatie op in vergelijking met interviews. Deelnemers stimuleren elkaar vaak en komen samen tot verschillende ideeën . Om een workshop zorgvuldig te laten verlopen, wordt er van te voren een plan bedacht welke onderwerpen aan bod komen en welke deelnemers er nodig zijn.

Bij observeren is het doel om het gedrag van het bedrijf te bestuderen. Hierbij wordt in de gehele periode in de gaten gehouden hoe de medewerkers werken en welke problemen ze tegenkomen met het werken van de huidige methode.

4. Planning, projectactiviteiten & resultaten

Hieronder is de planning opgesteld met alle projectactiviteiten en de resultaten die worden opgeleverd tijdens de uitvoering van het project. Activiteiten in deze periode kunnen nog worden gewijzigd of aangepast. Deze planning wordt dagelijks up-to-date gehouden met de mogelijke wijzigingen in het project.

Planning

	Week	eind Datum	Activiteiten	Resultaten
Fase 1:Probleem orientatie	1	12-Feb	Opstellen van plan van aanpak	
			Stakeholderanalyse	
			Contextanalyse: bedrijf, omgeving en strategie leren kennen	
			Overige hoofdstukken PvA	
			Business Canvas	
			Interview over probleemstelling met opdrachtgever/sales	
			Onderzoeksvragen opstellen	
			Zoektermen opstellen van het literatuuronderzoek	
			Risico's in kaart brengen	Eerste concept versie plan van aanpak
	2	19-Feb	Feedback verwerken van het plan van aanpak	Final versie Plan van aanpak
Fase 2:Measure			Huidige situatie in kaart brengen:	
			Identificeren van de problemen en de gevolgen hiervan bepalen	
			Root cause analyse en begrijpen van de oorzaken	
			Analyse van Eventum systeem	
			Verdiepen in de werkwijze, activiteiten en aanpak van zowel de ontwikkelaars als van de BPM specialisten	
			Interviews voorbereiden	
			Processen modelleren van betrokkenen in het softwareontwikkelingsproces(structuur)	Procesmodellen
	3	26-Feb	Verdiepende literatuuronderzoek uitvoeren over verschillende:	
			softwareontwikkelingsmethodes	
			software process improvement	

			implementatie van de nieuwe werkwijzen in de ict	
			Fieldresearsch over hoe andere organisaties binnen Nederland het software ontwikkelingsproces uitvoeren met behulp van Enquêtes(als tijd over is interviews/workshops)	Onderzoeksdocument
			Interviews afnemen met stakeholders	
			Eventueel onderzoeken naar andere tooling om te gebruiken	
			Huidige situatie vastleggen in rapport en presenteren van de huidige situatie voor relevante stakeholders	Rapport huidige situatie
		4	4-Mar	Optioneel: workshops/interviews met de klanten van B&P over de huidige situatie van het proces
				Gewenste situatie in kaart brengen en ontwerpen
				Interviews en workshops houden met betrokkenen(inclusief klanten indien mogelijk) van het software ontwikkelingsproces om meer informatie te verzamelen over hoe zij een effectieve samenwerkingsverband willen zien. Interview verslagen
				Gewenst situatie van het RFC systeem
				Gebruik maken van Empathy map/Customer Journey of andere technieken die betrekking hebben op processen en de mens
				Requirements opstellen over de gewenste situatie, deze vervolgens afstemmen met de stakeholders
		5	11-Mar	Gewenste situatie ontwerpen op basis van de opgestelde eisen/wensen van de betrokkenen in het proces en relevante technieken (oa BPM, human centered design)
				Gewenste situatie vastleggen in rapport en presenteren huidige situatie voor relevante stakeholders Rapport van de gewenste situatie
Fase 3: Analyse	6 t/m 8		1-Apr	Implementatie opstellen
				Analyseren van de gap tussen de huidige situatie en gewenste situatie
				Maak een plan om de gap “te dichten” met behulp van de verzamelde informatie
				Ontwikkel met de verzamelde informatie van de bovengenoemde werkzaamheden een nieuw voorstel uit.

			Workshop met stakeholders	
			Identificeer de Quick Wins om in de laatste fase door te voeren:	
			Maak hierbij een stappenplan	
			Presenteer nieuw implemtatieplan bij betrokkenen(eventueel ook de klant), verwerk vervolgens gekregen feedback uit tot een definitief verbetervoorstel	Implementatieplan
Fase 4: Improve	9 t/m 14	13-May	Uitvoeren van implementatieactiviteiten voor de opgestelde Quick Wins van het verbetervoorstel	
			Help hierbij zowel de klant als de collega's van Builders en Performers met de nieuwe werkwijze	
			Maak indien nodig ook handleidingen en gebruiksaanwijzingen	Handleidingen /gebruiksaanwijzingen
			Observeer of de Quick Wins ook werkelijk bij de betrokkenen geaccepteerd en gebruikt gaan worden	
			Meten en analyseren van de resultaten van de uitgevoerde Quick Wins.	
Fase 5: Control	14 t/m 17		Afronden bovengenoemde activiteiten:	
			Verbeter na de uitvoering van de Quick Wins het implementatieplan op basis van nieuwe inzichten	
			Documenteer een handleiding om de resterende implementatieplan overige verbetervoorstellen door een collega van Builders & Performers door te voeren	
			Afronden van de scriptie	sciptie

5. Kwaliteit & Scope

Planning

Zoals in de projectaanpak stond beschreven, wordt er op de methode van DMAIC gewerkt (van Vliet, 2014). Dit heeft als voordeel dat alle onderdelen op gestructureerde wijze aan bod komen in het project. Om uitloop van het project te voorkomen, is er een planning gemaakt in excel waarin wordt bijgehouden welke activiteiten in welke week wordt uitgevoerd. De planning wordt dan ook dagelijks bijgewerkt.

Informatie

Niet alleen tijdens de interviews/workshop wordt relevante informatie verzameld in het project, maar ook tijdens het meedraaien in het team, zoals het bijwonen van vergaderingen of andere besprekingen/activiteiten komt bruikbare informatie beschikbaar voor het project. Om te zorgen dat er geen informatie verloren gaat, wordt er een logboek bijgehouden met daarin alle uitgevoerde activiteiten en observaties. Deze informatie wordt ook gebruikt om de huidige- en gewenste situatie vast te leggen.

Interviews

In het project worden ook regelmatig interviews/workshops gehouden. Om te zorgen dat er genoeg informatie wordt verzameld uit deze interviews, is een strategie belangrijk. Allereerst worden de interviews ruim op tijd gepland. De medewerkers of andere betrokkenen hebben niet altijd ruimte voor interviews, daarom is het belangrijk de interviews op tijd te plannen. In de voorbereiding voor een interview wordt er een interviewboekje gemaakt. De vragen in het interview zijn voornamelijk open-vragen, dit zorgt voor veel meer informatie ten opzichte van gesloten vragen.

Onderzoek

Voor het onderzoek van het project wordt er ook een literatuuronderzoek uitgevoerd naar de verschillende softwareontwikkelingsprocessen. Om te zorgen dat de gezochte informatie bruikbaar is, wordt er voornamelijk gezocht in Google Scholar en de databanken van de Haagse Hogeschool en de Hogeschool van Rotterdam. Ook worden de bronnen gecontroleerd op actualiteit, bekendheid van de auteur en organisatie en de opgesomde bronnen die het artikel ondersteunen.

Om een onderzoek betrouwbaar uit te voeren, moet er in alle tijden aan gedacht worden dat ik; objectief en onafhankelijk blijf, controleerbaar en toetsbaar ben en nauwkeurig te werk ga.

5.1 Scope

Voor het project wordt als eindproduct een implementatieplan opgeleverd van het uitgevoerde onderzoek. Het implementatieplan omvat een plan om de gap tussen de huidige situatie en de gewenste situatie te dichten. Daarin wordt een stappenplan beschreven om de veranderingen door te voeren. Hoewel de oplossingen van het project nog niet bekend zijn, wordt er gekeken welke oplossingen als Quick wins kunnen worden doorgevoerd.

Waarschijnlijk kunnen niet alle oplossingen worden doorgevoerd i.v.m. met de beschikbare tijd van het project. Er wordt wel voor de overige oplossingen een document beschreven om ze op een later termijn alsnog door te voeren.

6. Stakeholders

Voor dit project hebben verschillende stakeholders van zowel binnen als buiten de organisatie invloed op het project. Om inzicht te krijgen over de verschillende belangen van de stakeholders is er een stakeholdersanalyse gemaakt. Hierbij is bekeken wat de belangen en interesse van de stakeholders zijn. De methodes die hieronder zijn gebruikt, komt van het boek *" De kunst van veranderen"* (de Witte & Jonker, 2014).

Stakeholders analyse

Directe stakeholders	Belang	Emotie	Invloed (1/5)
Opdrachtgever	Het belang van de opdrachtgever is dat zijn organisatie efficiënter gaat werken. Hij hoopt dat de organisatie na dit project een stap volwasser is.		5
Sales	Zijn belang is dat er met de nieuwe methode betere software kan worden aangeboden aan de klant.		2
BPM specialist	Zijn belang is dat de communicatiestroom tussen de ontwikkelaar en de klant efficiënter verloopt.		3
Klant	De klant heeft het belang dat zijn eis/wens van het systeem efficiënter wordt afgehandeld.		1
Ontwikkelaar	Het belang van de ontwikkelaars is dat ze zich beter thuis voelen in de methode/werkwijze.		Stefan: 4 Menno: 3 Harry: 3 Cindy: 2 Michael: 1

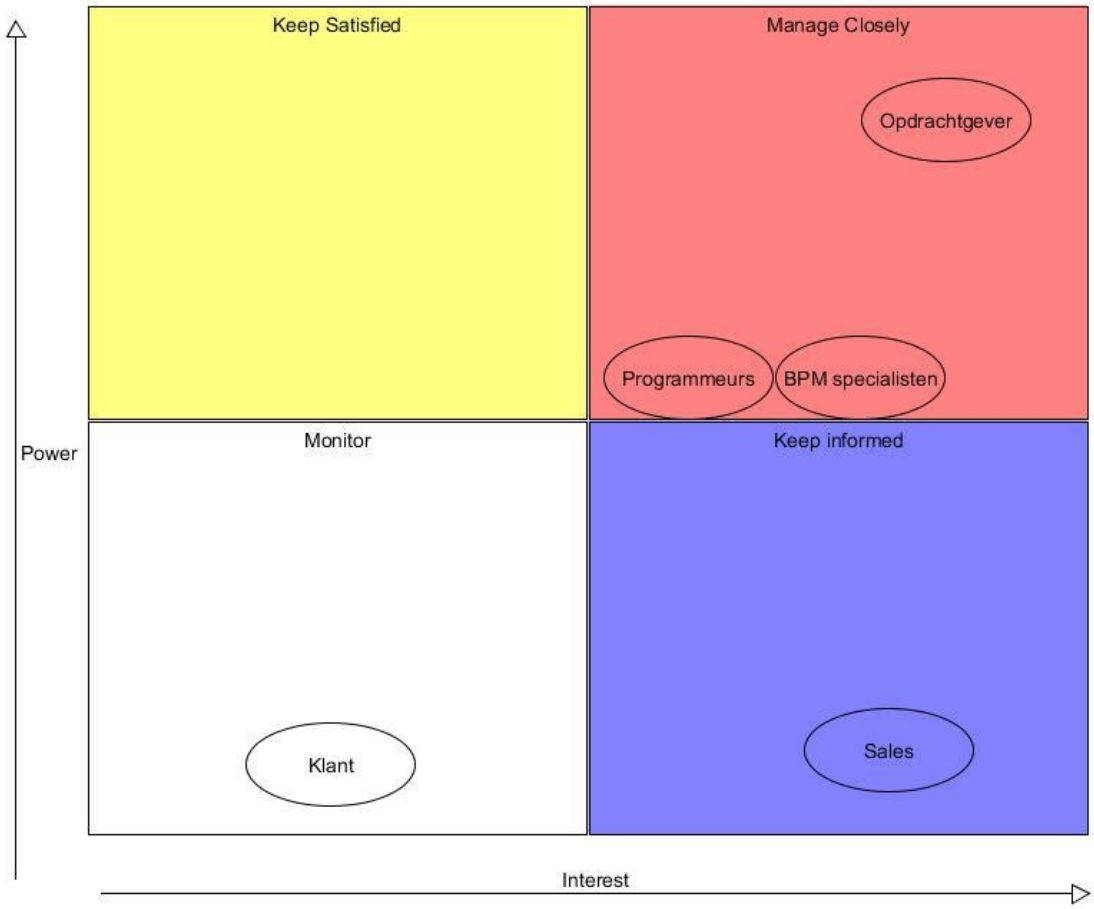
Hieronder is een stakeholdersmatrix gemaakt om te zorgen dat er strategisch wordt omgegaan met de verschillende stakeholders van het project. Van beneden naar boven in de tabel wordt aangegeven hoeveel macht ze hebben over het project. Van links naar rechts wordt aangegeven hoe geïnteresseerd de stakeholders zijn met het project. Hieruit ontstaan vier bouwstenen:

Monitor: Zij hebben weinig macht en interesse in het project, daarom hoeven ze alleen in de gaten worden gehouden.

Keep Informed: Zij hebben weinig macht, maar zijn wel erg geïnteresseerd in het veranderingsproces. Daarom is het belangrijk deze groep te informeren en misschien te betrekken in het project.

Keep Satisfied: Zij hebben wel macht, maar zijn minder geïnteresseerd in het project. Deze groep moet alleen tevreden worden gehouden tijdens de verandering.

Manage Closely: Zij hebben veel macht en zijn ook erg geïnteresseerd in het project. Deze groep is erg belangrijk om te betrekken in het project. Zij worden betrokken voor alle relevante beslissingsmomenten en resultaten van het veranderingstraject.



De klant wordt in dit project vooral gemonitord. De interesse van de klant in dit project is niet erg groot, omdat zij het niet relevant vinden hoe het interne proces bij builders & performers verloopt. Voor de klant is het alleen belangrijk dat de eisen/wensen efficiënt worden afgehandeld.

Sales is ook zeer geïnteresseerd in het project. Zij zijn degene die het product moeten gaan aanbieden bij de klant. Zij gaan hoogstwaarschijnlijk niet veel werken met de nieuwe methode, daarom zullen ze ook niet veel macht hebben over het project.

Voor de programmeurs, BPM-specialist en de opdrachtgever is het belang groot voor het project. Zij zijn degene die het meest betrokken worden in de nieuwe methode. Belangrijk is om deze groep in het gehele traject te betrekken. Voor elke verandering en resultaat wordt er met deze groep regelmatig gecommuniceerd. Ook worden zij betrokken in alle cruciale beslissingen van het project.

7. Randvoorwaarden en risico's

Een aantal randvoorwaarden zijn opgesteld om het project succesvol te laten verlopen:

- Eigen werkplek met laptop
- Tijd om de medewerkers van de organisatie te interviewen, workshops te houden en te betrekken in de Quick Wins van de nieuwe methode
    - Eventueel ook tijd van klanten
- Tijd van de begeleider voor een wekelijks gesprek over het verloop van het project

De volgende risico's kunnen een rol spelen bij het dit project:

Risico	Maatregel	Kans(1/5)	Impact(1/5)
Ziekte van de medewerkers	Op tijd beginnen met interviews en flexibel te werk gaan.	2	1
Project is niet haalbaar binnen de tijd(deadlines worden niet gehaald)	Elke week controleren of de opgestelde planning wordt nageleefd. Mochten de activiteiten niet binnen de planning passen, dan moeten de activiteiten worden geprioriteerd. Vooral het onderdeel waaruit de quick wins worden gedefinieerd, is het belangrijk goed te beoordelen hoeveel tijd ervoor nodig is.	3	4
Medewerkers/cultuur van de organisatie zijn niet bereid om te werken met een de nieuwe methode.	De medewerkers informeren wat de voordelen/baten zijn om te willen werken met de nieuwe methode. Daarnaast, zoals het boek "de kunst van veranderen" vermeld, voornamelijk richten op de groep die wel bereid is om mee te werken in de verandering. De kans is dan groter dat dan de andere groep ook meegaat in de nieuwe veranderingen.	2	4
Het eindproduct(implementatieplan) sluit niet aan op de eisen/wensen van de opdrachtgever/organisatie	Zorg voor meerdere iteraties in het project. Dit zorgt ervoor dat de opdrachtgever niet voor verrassingen komt te staan.	2	4

Kennis niet voldoende van verschillende softwareontwikkelingsmethodes	Extra tijd investeren in het onderzoeken van de methodes die je nog niet kent, dan de methodes die je wel kent.	5	2

8. Communicatie

Het is belangrijk dat de communicatiestroom tussen het project en de belanghebbende goed verloopt. Anders krijg je het risico's dat de betrokkenen voor verrassingen komen te staan. Voor het project wordt er met de volgende contactpersonen gecommuniceerd:

Ronny Randham(opdrachtgever)

De opdrachtgever van het project wordt wekelijks up-to date gehouden met een mondelinge overleg. In het overleg wordt op een scrumachtige wijze verteld wat er is gedaan, wat er gaat gebeuren en tegen welke problemen wordt aangelopen. Als er in de week ook resultaten worden opgeleverd, worden deze besproken met de opdrachtgever.

Henk Hoeksema(begeleider)

Met mijn begeleider wordt dagelijks overlegd over de uitgevoerde activiteiten van het project. Elke week wordt er met de begeleider besproken of de planning nog wordt behaald en de eventuele problemen die een rol (kunnen gaan) spelen in het project.

Het team

Aan het eind van elke fase worden verschillende resultaten opgeleverd. De resultaten worden vervolgens voor het team van Builders & Performers gepresenteerd. Dit zorgt ervoor dat er bruikbare feedback wordt geleverd, waardoor het eindresultaat beter kan worden aangescherpt.

De klant

In het project wordt eventueel ook informatie verzameld van de klant. De klant heeft niet altijd tijd hiervoor, daarvoor is het belangrijk om ruim van te voren te communiceren voor een afspraak.

Roeland Loggen(examinator)

Roeland wordt één keer per week gemaild met een statusrapport. In het statusrapport wordt vermeld hoe de voortgang van de afgelopen weken is gegaan. Ook wordt er aangeven of er blokkades in de weg zitten. Daarnaast wordt er ook een logboek meegestuurd van de afgelopen week. Mocht er een noodsituatie voorkomen, dan wordt er direct contact genomen.

Halverwege de afstudeerstage wordt Roeland uitgenodigd voor een mondelinge afspraak om de voortgang samen met mijn begeleider te bespreken.

9. Bibliografie

de Witte, M., & Jonker, J. (2014). *De kunst van veranderen.* Deventer: Kluwer.

Fischer, T., & Julsing, M. (2014). *Onderzoek doen!* Groningen: Noordhoff Uitgevers.

Software development process. (2014, Augustus). *Software development process*. Opgehaald van Wikipedia: https://en.wikipedia.org/wiki/Software_development_process

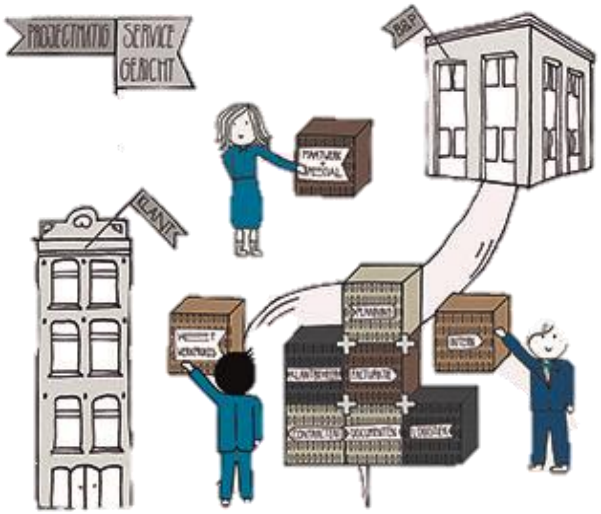
The BIG 6. (2014). *The Big6: A Good Place to Get Started*. Opgehaald van The BIG 6: http://big6.com/pages/kids/grades-k-6/articles-k-6/big6-a-good-way-to-get-started.php

van Veen, M., & Westerkamp, K. (2012). *Deskresearch.* Pearson Benelux: Amsterdam.

van Vliet, V. (2014, Juni 10). *DMAIC model(verbetermethode)*. Opgehaald van Toolshero: http://www.toolshero.nl/kwaliteitsmodellen/dmaic-model/

BUILDERS & PERFORMERS

HUIDIGE EN GEWENSTE SITUATIE



Auteur: Ruben van Stralen

Datum: maandag 30 mei 2016

Versie: 1.0

1 Inhoud

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

1 Inleiding

Met behulp van interviews en observaties is de huidige situatie en de gewenste situatie vastgelegd van het ontwikkelproces van B&P. Dit heeft tot doel om te analyseren welke onderdelen binnen de organisatie goed gaan en wat er verbeterd kan worden. In het tweede hoofdstuk is beschreven waarom er veranderd moet worden en welke stakeholders in deze verandering een rol spelen. In het derde hoofdstuk wordt verteld wat er speelt in de huidige situatie en wat de gewenste situatie is. Met de opdrachtgever zijn in het vierde hoofdstuk de business requirements opgesteld wat de verandering moet gaan opbrengen. Uit de interviews met de andere stakeholders zijn de gebruikersrequirements opgesteld. De requirements zijn door de stakeholders geprioriteerd. In de conclusie is beschreven op welke punten de organisatie kan veranderen.

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

2 Waarom veranderen?

In de eerste fase van het project is er een interview afgenomen met de BPM-er en een met de opdrachtgever/directeur van B&P. Het doel van het interview was om tot de rootcause van het werkelijke probleem te komen en waarom er veranderd moet worden. In de huidige probleemstelling wordt geformuleerd dat de samenwerking tussen de BPM-er en de ontwikkelaars niet efficiënt verloopt. Dit heeft als gevolg dat er veel rework plaatsvindt (bij de klant),frustraties zijn binnen het team en de kwaliteit van de software onvoldoende is. De gevolgen van dit probleem zorgen er uiteindelijk voor dat de organisatie niet of amper verder kan groeien. Daar moet verandering in worden gebracht.

Uit de interviews om tot de rootcause (bijlage B) van het probleem te komen zijn drie verschillende oplosrichtingen gekomen. Als eerste het huidige ontwikkelproces. Dat is het proces vanaf het moment dat een klant een eis/wens heeft tot aan de oplevering van de aanpassing. De medewerkers bij Builders & performers hebben ieder een eigen beeld van hoe het huidige proces/methodiek verloopt in de organisatie. Dit komt omdat er geen heldere afspraken zijn gemaakt. Vrijwel niks staat gedocumenteerd. Met andere woorden, er is geen heldere omschreven proces aanwezig in de organisatie. Dit heeft tot gevolg dat de samenwerkingsvormen, -lijnen en verantwoordelijke niet duidelijk aanwezig zijn.

Het tweede punt is dat een ontwikkelaar een slecht overzicht heeft welke taken hij/zij de komende periode gaat uitvoeren. In de organisatie wordt een tool “Eventum” gebruikt om alle eisen en wensen van klanten te registeren en te verwerken. Eventum heeft zijn beperkingen. Er is geen duidelijke planning aanwezig, waardoor de ontwikkelaar niet weet wat hem te wachten staat. Ook ontbreekt er relevante informatie in de RfC om de aanpassingen te bouwen. Kortom, de RfC tool Eventum moet gewijzigd en/of worden vervangen.

Het derde punt is dat er vaak iets anders wordt opgeleverd dan wordt gevraagd. Een oorzaak hiervan is dat de eisen/wensen van de klant anders worden geïnterpreteerd door de ontwikkelaar. Volgens de BPM-er is de oorzaak dat de ontwikkelaar niet doorvraagt en alleen wil gaan bouwen. Echter, wordt vanuit de ontwikkelaars een ander verhaal verteld wat hiervan de oorzaken zijn. Daarnaast speelt mee dat de klant vaak niet weet wat hij/zij wil hebben, omdat de klant nog geen goed beeld heeft van zijn eigen eis/wens.

Hoewel het momenteel bij de organisatie Builders & Performers nog goed draait, dreigt de organisatie in de toekomst in de problemen te komen. Dat is de reden dat er verandering in de organisatie moet gaan plaatst vinden.

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

De verandering heeft impact op zowel de klant, de organisatie, het team en het individu. De verandering gaan zorgen voor:

De klant:	Betere klantwaarde propositie, dit betekent betere kwaliteit van de opgeleverde software en minder rework.
De organisatie:	Heeft positieve gevolgen voor de financiële baten en de groei van het bedrijf.
Het team:	Heeft belang dat er efficiënter wordt gewerkt in het algemene proces en een betere samenwerking tussen de BPM-er en de ontwikkelaars. Ook gaat de verandering zorgen voor minder frustraties binnen het team en voor een betere werkomgeving.
Individu:	Gaat zich beter thuis voelen binnen het proces. De verandering zorgt voor meer verantwoordelijkheid bij het individu.

Zoals in de titel van dit hoofdstuk staat aangegeven, waarom moet de organisatie eigenlijk veranderen? Omdat het organisatorisch nog niet goed loopt, en vooral de samenwerking van de BPM-er en de ontwikkelaar, maar ook de samenwerking onderling. Builders & Performers staat goed in de markt, organisatorisch kan het beter. De verandering moet zorgen om een stap hoger te komen.

Builders & Performers speelt momenteel nog in de "Jupiler League", maar met de toekomstige veranderingen gaan we met het gehele team strijden in de Eredivisie".

Dat is de visie waarop deze verandering naar toe streeft.

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

3 Huidige en gewenste situatie

In dit hoofdstuk wordt de huidige en gewenste situatie in kaart gebracht van de opgestelde probleemstelling. De huidige en gewenste situaties zijn beschreven met informatie uit interviews met de BPM-er, ontwikkelaars en de opdrachtgever. Daarnaast is informatie gebruikt uit eigen observatie gericht op de interne communicatie, opgestelde documentatie, werkwijze en de RFC tool Eventum. De situaties zijn onderverdeeld in de vier bouwstenen:

Structuur:	Bestaat uit de primaire en secundaire processen en geordende activiteiten.
Technologie:	Hardware/software en andere middelen/gereedschap om processen uit te voeren.
Medewerkers:	Competenties en individuele kenmerken van medewerkers.
Cultuur	Zo zijn onze manieren, collectieve gedragspatronen, zowel formeel als informeel.

In **bijlage A** zijn alle opmerkingen verdeeld in vier verschillende kleuren. Elke kleur is van een verschillende belanghebbende van het project.

Oranje: **Ontwikkelaars**

Blauw: **BPM-er**

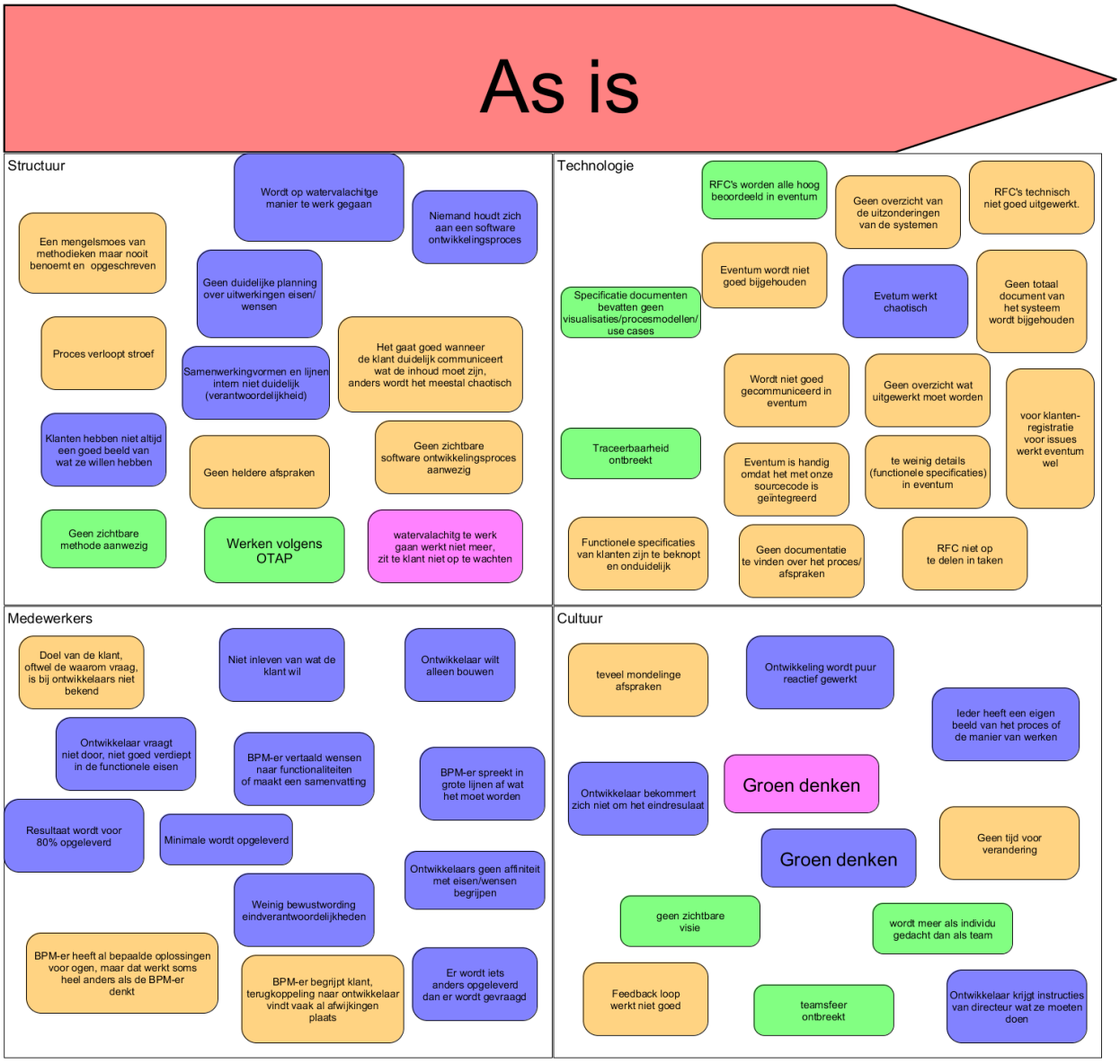
Groen: **Observator**

Roze: **Algemeen Directeur**

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

3.1 Huidige situatie(As is)

In de huidige situaties is met informatie uit interviews en observaties beschreven hoe het huidige bedrijfsproces wordt uitgevoerd binnen de organisatie en tegen welke problemen ze aanlopen .



Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

Structuur

In de bouwsteen structuur is voornamelijk gekeken hoe ze procesmatig en methodisch werken binnen de organisatie. Uit het interview met de opdrachtgever is naar voren gekomen dat er wel volgens een methode wordt gewerkt, alleen is in het bewustzijn binnen de organisatie er niet. Momenteel worden projecten bij verschillende klanten op andere manieren aangepakt. Bij klanten die precies kunnen benoemen wat de eisen en wensen van het aanpassing(en) zijn, wordt er een watervalachtig methode gebruikt. Er wordt een contract met de klant opgesteld waarin alle functionele specificaties staan. Dit document wordt vervolgens aangeleverd bij de ontwikkelaars, die aan de hand daarvan de aanpassingen bouwen(zie bijlage C en D).

Klanten die van te voren nog niet goed wisten wat ze precies wilden, wordt op een iteratieve manier gewerkt. Bij klanten die wijzigingen sturen aan het bestaande systeem wordt agile achtig manier te werk gegaan.

Projectmatig is er geen vast omlijnde aanpak om een product of een productwijziging op te leveren bij de klant. Dit is ook uit de andere interviews naar voren gekomen. Sommige ontwikkelaars hebben het idee dat er een mengelmoes van methodes wordt gebruikt, maar eigenlijk is dat nooit benoemt of opgeschreven. Anderen dachten dat er helemaal geen methode wordt gehanteerd binnen de organisatie. Dat is een van de redenen dat niemand zich houdt aan een softwareontwikkelingsproces. Met andere woorden, het proces loopt stroef en kan duidelijk worden verbeterd.

Een ander probleem dat vaak voorkomt, is dat de klant iets anders krijgt opgeleverd dan werd verwacht. Of er ontbreken bepaalde onderdelen bij de oplevering van het systeem. Een mogelijke oorzaak daarvan is dat de klant niet altijd van te voren voor ogen had wat ze precies verwachten van het ontwikkelde informatiesysteem. Een andere mogelijke oorzaak is dat de doorgestuurde eisen/wensen te beknopt beschreven zijn, waardoor de ontwikkelaar een eigen interpretatie heeft van wat het werkelijk moet worden. Er zijn voorbeelden benoemt bij klanten waar het wel goed is gegaan, doordat zij van te voren duidelijk hadden gecommuniceerd waarom, wat en hoe ze het wilden hebben. De klanten die nog niet precies wisten waarom, wat en hoe, wordt meestal als een chaotisch project gezien bij B&P.

Een overzicht van de planning, de uitgevoerde werkzaamheden en door wie misten ook in het proces. Hoewel daar wel een tool genaamd Eventum voor wordt gebruikt, schiet de tool te kort als het gaat om projectmatig werken.

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

Technologie
Zoals hierboven al werd verteld, is Eventum niet ideaal bij sommige onderdelen in het ontwikkel proces, daarnaast wordt de tool ook niet actief gebruikt binnen het proces. Hieronder is een lijst opgesomd met eigen bevindingen van de tool:

- Elke RFC wordt beoordeeld met een prioriteit. De prioriteit moet aangeven hoe urgent de ingekomen RFC is, waardoor kan worden bepaald welke RFC's voorrang hebben voor de volgende release. Nu is het opgevallen dat tijdens elke release de RFC's allemaal high of critical worden beoordeeld. De oorzaak hiervan is dat de klant zijn eis/wens naar eigen mening erg belangrijk vindt. Doordat alle RFC's hoog worden beoordeeld, verliest de prioriteit zijn waarde. Hierdoor kan men niet goed beoordelen welke RFC's relevanter zijn om eerst uit te voeren en te implementeren en welke RFC's minder belangrijk zijn.

- Relevante informatie over de voortgang van het proces ontbreekt vaak. Als een RFC binnenkomt van de klant, wordt alleen de eis en wens beschreven in het proces en een reactie hoeveel de tijdinvestering voor de RFC's nodig is. Voor de eis/ wens wordt niet altijd aangeven of:

    - Het wordt begrepen
    - Er al een oplossing voor is
    - dat de oplossing nog bedacht moet worden
    - de tevredenheid van de klant na de implementatie

  Oftewel, er is amper een zichtbare structuur te vinden vanaf het begin dat een RFC binnenkomt tot de oplevering van de RFC.

- De RFC van de klant wordt niet op een consistente wijze beschreven. Hierdoor is het voor gebruikers van het systeem niet snel overzichtelijk wat er wordt gevraagd met de eis/wens van de klant.

- De kans op impact van de RFC op andere bestaande processen wordt niet aangegeven. Een eis of wens kan impact hebben, namelijk op de andere processen van het systeem of de organisatie. Door de impact niet op te nemen, is er een verhoogd risico dat er iets mis kan gaan tijdens en na de oplevering.

- In sommige RFC's zitten meerdere requirements van de klant. Hierdoor is het minder inzichtelijk hoeveel eisen en wensen daadwerkelijk worden doorgevoerd. De RFC is vervolgens niet op te delen in taken.

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

- Het plannings scherm van Eventum is niet erg overzichtelijk. Het is lastig om in het systeem te zien hoe de RFC's in welke volgorde worden uitgevoerd.
- Per RFC wordt bepaald hoeveel tijd er nodig is om de eis/wens uit te voeren. Er wordt zelden beschreven waar de tijd aan wordt besteed. Vaak wordt er ingeschat hoeveel tijd er nodig is om de eis/wens te ontwikkelen, maar ontbreekt de berekening van de overige activiteiten voor het afhandelen van de RFC(bijvoorbeeld de communicatie met klant).
- Wordt niet ondersteund om visualisaties, procesmodellen of use cases toe te voegen

Zoals door de medewerkers wordt aangegeven: de tool werkt chaotisch en wordt niet actief en communicatief gebruikt. De tool heeft ook zijn pluspunten, het is namelijk ideaal om de issues van klanten te registreren en te verwerken en is geïntegreerd met de sourcecode.

Verder is het vastleggen, wijzigen en managen van informatie een fundamenteel probleem. Hierdoor is er amper overzicht bij de medewerkers. Onder informatie wordt verstaan:

- Processen en afspraken
- Technische documentatie, vooral uitzonderingen op het systeem
- Beknopte en onduidelijke eisen/wensen van klanten, er missen zowel functionele als technische details
- Geen totaal document van een opgebouwd systeem bij een klant
- Informatie is niet traceerbaar

Volgens de interviews zijn de bovengenoemde punten een fundamentele oorzaak dat er veel rework bij klant plaats vindt.

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

Medewerker

Bij de bouwsteen medewerker is het begrijpen van de eisen en wensen van de klant een essentieel onderwerp. Bij de klant wordt namelijk niet altijd datgene opgeleverd wat de klant voor ogen had, waardoor er onnodig rework plaatsvindt. In de meeste gevallen wordt er in de huidige situatie voor 80% opgeleverd aan de klant en is er 20% rework nodig om het geheel compleet e maken . Over dit onderwerp heeft zowel de BPM-er als de ontwikkelaar een eigen verhaal “het waarom” de eis/wens verkeerd wordt geïnterpreteerd.

De BPM-er spreekt in grote lijnen met de klant af wat er moet worden gebouwd. Een document wordt opgesteld met de functionele specificaties of een samenvatting daarvan. Dit wordt vervolgens aangeleverd bij de ontwikkelaar. Volgens de BPM-er moet de ontwikkelaar als de requirement niet wordt begrepen doorvragen. Dit wordt momenteel niet (altijd) gedaan, omdat de ontwikkelaar weinig affiniteit heeft om zich in te leven in de klantwens en alleen wil bouwen.

Aan de andere kant wordt de opgeleverde informatie van de BPM-er aan de ontwikkelaar te beknopt ervaren. De functionele specificaties of RFC’s van de klant bevat te weinig details. Voor de ontwikkelaar is wel duidelijk de “wat” van de requirement, maar niet duidelijk de *“waarom”* en *“hoe”* van de requirement. Ook wordt er bij de BPM-er al in oplossingen gedacht voor de klant, maar dat werkt soms heel anders in het informatiesysteem.

Cultuur

Een niet zichtbare software ontwikkelingsproces zorgt voor verwarring bij de medewerkers van B&P. Momenteel heeft ieder een eigen perspectief van de huidige werkwijze. Er worden vaak nog veel mondelinge afspraken gemaakt, en niks wordt op papier vastgelegd. Op communicatieniveau gaat het dan ook weleens verkeerd.

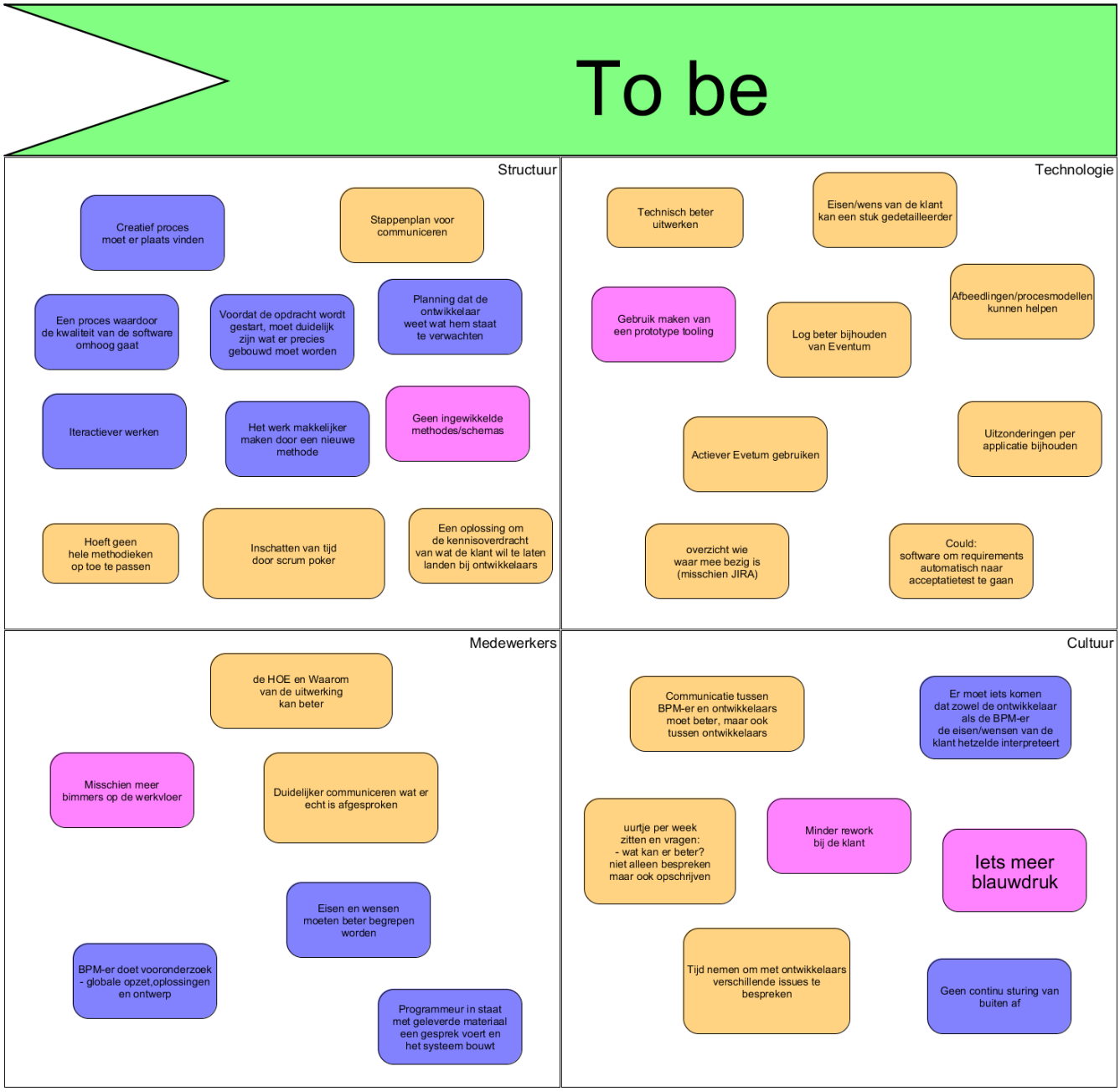
Hoewel de directeur communiceert met de ontwikkelaar wat ze moeten doen of wat ze aan het doen zijn, verloopt de communicatie met de ontwikkelaars onderling stroef. Er wordt bijvoorbeeld amper feedbackmomenten gehouden na een oplevering en afspraken zijn niet duidelijk waardoor er ook weinig bewustwording is wie de rol heeft als eindverantwoordelijke.

De bovengenoemde problemen werden bij verschillende medewerkers ervaren. Ook werden hiervoor al verschillende oplossingen bedacht zoals een scrummeeting, scrum poker of gaan werken met de softwaretool JIRA. Deze oplossingen zijn nooit binnen de organisatie geïmplementeerd, omdat er amper tijd werd genomen om de veranderingen door te voeren.

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

3.2 Gewenste situatie(to be)

In de gewenste situatie is beschreven hoe de medewerkers van B&P het graag anders zouden zien.



Structuur

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

In de huidige situatie is er geen heldere software ontwikkelingsproces aanwezig. Dit kan volgens de medewerkers anders. Hoewel er geen ingewikkelde of gehele methode hoeft te komen, moet er wel een methode/werkwijze plaats vinden om de kwaliteit te verhogen van de software. De opdrachtgever wil dat een methode zich ook richt op de gedachtegang van Agile, omdat de markt daar steeds meer om vraagt[1]. De methode moet gaan zorgen voor:

- Een stappenplan om het proces gestructureerd uit te voeren
- Een handvat met beschreven werkwijze agile
- Betere kennisoverdracht van wat de klant wil
- Inschatten van de RFC op tijd
- Duidelijke planning
- Plaats vinden van een creatief proces
- Interactiever werken

Technologie

Eventum wordt momenteel nog niet optimaal gebruikt, dit wordt ook erkend door de medewerkers en willen hier ook verandering in. De tool moet actiever worden gebruikt en het log kan beter worden bijgehouden. Verder heeft de tool Eventum ook zijn beperkingen, een fundamentele beperking is het overzicht en planning van de RFC's die uitgevoerd gaan worden. De tool ondersteund dit niet. Er wordt daarom gekeken om naar een andere tooling te zoeken, als voorbeeld werd JIRA genoemd.

Daarnaast komt de vraag om beter om te gaan met het verwerken en managen van informatie. Volgens de ontwikkelaars kan de eis/wens een stuk gedetailleerder, hierbij zouden procesmodellen, schermafbeeldingen en prototypes een bijdrage kunnen leveren. Ook vanuit technische oogpunt kan de uitwerking beter van de ontwikkelaars. Het bijhouden van de uitzonderingen per applicatie kan op langer termijn een hoop tijd besparen voor de ontwikkelaar. Momenteel moet de ontwikkelaar bij elke wijziging gaan uitzoeken op welke onderdelen de requirement impact heeft. Er moet een beter hulpmiddel komen om de functionele specificaties en technische uitzonderingen te traceren en te managen.

Medewerkers

[1] Nathan, W., & Holte, M. (2016, Januari 29). *The End of the Waterfall as We Know It*. Retrieved from Gartner: http://www.gartner.com/document/3188962?ref=solrAll&refval=163874692&qid=e1c8f00d8e5fb3f454056c1531393612

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

De requirements van de klant moet in de nieuwe situatie beter worden begrepen. De droomsituatie van de BPM-er is, dat de BPM-er bij de klant een vooronderzoek doet, zet een globale beschrijving met oplossingen en een ontwerp. Met het geleverde materiaal hoopt de BPM-er dat de ontwikkelaars in staat zijn om een gesprek te houden met de klant om vervolgens het systeem te bouwen.

De ontwikkelaars zien het proces graag anders, zoals wordt benoemd hebben zij de wens voor gedetailleerdere eisen/wensen van de BPM-er. Hierin wordt dan niet alleen de “wat” beschreven van de klant, maar ook de “waarom” en “hoe” van de klant. Volgens de directeur is het dan misschien ook wel een idee om meer BPM-ers op de werkvloer te zetten.

De gewenste situatie voor het realiseren van de requirements van de klant heeft twee kanten;

- of er wordt meer aandacht besteed aan de vooranalyse bij de klant, waarmee een functioneel ontwerp wordt opgeleverd met gedetail-leerde requirements en bijhorende objecten.

- Of de BPM-er maakt een globale beschrijving en aan de hand daar-van wordt verwacht dat de ontwikkelaar meer interactie heeft met de klant om zijn eis/wens beter te begrijpen.

Cultuur

Op het gebied van communicatie moet het beter in de organisatie, hierdoor voorkom je frustraties binnen het team en zal er uiteindelijk minder rework plaats vinden. Volgens de ontwikkelaars kan dit worden behaald als er meer met elkaar wordt gecommuniceerd. Dat betekent dat de BPM-er en ontwikkelaars gezamenlijk de eisen en wensen bespreekt, zodat beide partijen de eis en wens van de klant hetzelfde interpreteert. Tussen de ontwikkelaars kan er ook meer gecommuniceerd worden om bijvoorbeeld verschillende issues gezamenlijk te bespreken of eens per week te zitten om te spreken wat er goed ging en wat er beter kan.

Uiteindelijk moet er cultuur ontstaan dat het gehele team van B&P meer en beter met elkaar communiceert, dus geen sturing telkens van buitenaf, maar dat ieder verantwoordelijkheid neemt om samen verschillende onderdelen in het proces te bespreken.

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

4 Requirements

Uit de interviews en observaties zijn de business- en gebruikersrequirements vastgelegd. De requirements zijn met alle betrokkenen gevalideerd en geprioriteerd. Hieronder zijn de requirements opgesomd wat veranderd moet worden:

Business requirements:

- B1: De organisatie wil een methode/werkwijze methode invoeren dat consistent het bedrijfsproces ondersteund en optimaliseert.
- B2: De organisatie wil in staat zijn 30% meer werk op te leveren binnen dezelfde tijd.
- B3: De organisatie wil dat de uitvoering 25% efficiënter wordt verwerkt.
- B4: De organisatie wil zorgen dat er minder frustraties plaats vinden binnen het team.
- B5: De organisatie wil dat er 40% minder rework plaats vindt.
- B6: De organisatie wil dat de effectiviteit van de uitvoering omhoog gaat.
- B7: De opdrachtgever wil een transparante methode die gebruik maken van de methodes en technieken van agile en waterval.

Gebruikers requirement:

- G1: De ontwikkelaar wil een gedetailleerdere uitwerking in een functioneel ontwerp van de eis & wens bij klanten.
- G2: De ontwikkelaar wil een gedetailleerdere uitwerking van de RFC bij bestaande klanten.
- G3: De ontwikkelaar wil dat de technische uitzonderingen en functionele specificaties aan een opleverde systeem worden bijgehouden.
- G4: Het team van B&P wil een beter overzicht van de planning van de verschillende projecten.
- G5: De ontwikkelaar wil dat er meer met het team wordt gecommuniceerd.
- G6: De ontwikkelaar wil dat het inschatten van een RFC op tijd gezamenlijk verloopt.
- G7: De ontwikkelaar wil een beter communicatieplan tussen de BPMer en de ontwikkelaars.

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

- G8: De BPM-er wil dat er meer bewustwording bij de ontwikkelaar komt als eindverantwoordelijke in het ontwikkelproces.
- G9: De BPM-er wil dat de ontwikkelaar meer zelfstandige interactie heeft met de klant wanneer het proces dit vraagt.
- G10: Het team van B&P wil heldere afspraken en structuur.

4.1 Prioriteren requirements

Alle requirements zijn door de gerelateerde stakeholders beoordeeld. Dit werd gedaan in een enquête waar je per requirements op een schaal van 1 tot 10 kon aangeven hoe belangrijk het probleem speelt en hoe urgent de requirement is. De volgende formule is gebruikt om de waarde van de requirement uit te rekenen:

Formule = Urgent + Belangrijk / 2

In de onderstaande tabel is de waarde van de requirement aangeven:

Business requirements	
B1	8.25
B2	7.5
B3	8
B4	8
B5	7.5
B6	7.75
B7	7.375

Gebruikersrequirements	
G1	7.125
G2	7.375
G3	7.75
G4	6.916667
G5	7.125
G6	6.916667
G7	7.375
G8	8
G9	7
G10	7.25

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

5 Conclusie

Uit de interviews en observaties is geconstateerd dat er bij Builders & Performers geen zichtbare methode aanwezig is. Binnen de organisatie is er niet een framework/methode vastgesteld die gehanteerd moet worden. Vanuit de medewerkers en de organisatie komt daarom duidelijk de vraag om te willen werken met een softwareontwikkeling methode. De methode moet structuur bieden in de organisatie, dat van iedere medewerker zijn werkwijze en verantwoordelijkheden bekend is en wat er opgeleverd moet worden qua artifacts. In het onderzoeksdocument is onderzocht hoe en welke methode kan worden gebruikt voor B&P.

Een andere punt om te verbeteren is de uitwerkingen van de requirements en RfC's . Deze waren voorzien van te weinig informatie, dit had tot gevolg dat er relatief veel rework plaatst vindt bij de klant. Er moet in de verbeterende methode worden gekeken welke activiteiten en artifacts er nodig zijn om de eisen en wensen beter over te brengen naar zowel de klant als naar de ontwikkelaar. In het onderzoeksverslag is te lezen hoe deze activiteit kan worden verbeterd.

Het laatste onderdeel waar naar gekeken kan worden is de tooling. Momenteel worden er bij Builders & Performers amper tot geen hulpmiddelen gebruikt dat zorgt voor:

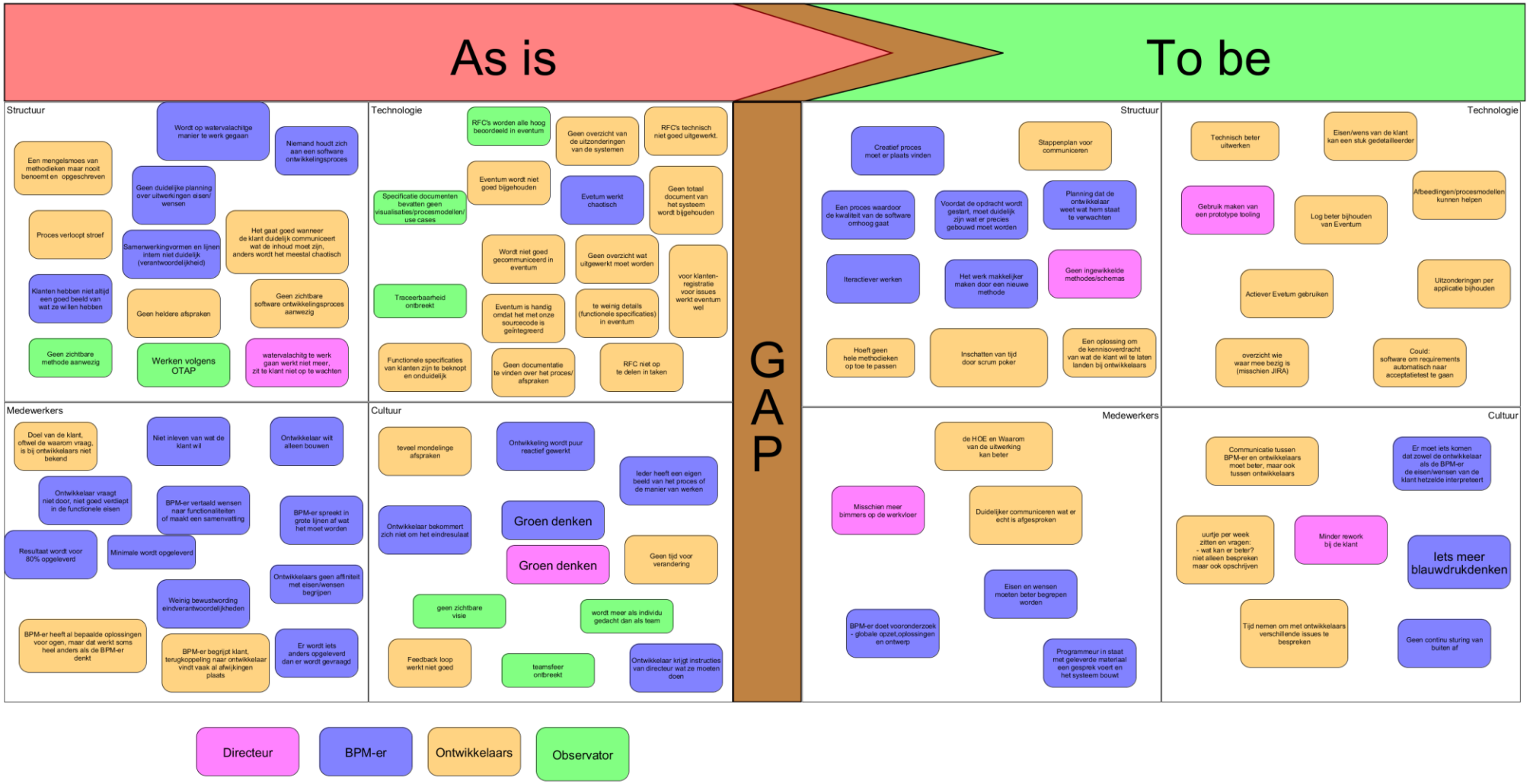
- Een overzicht van de planning.
- Tooling waarmee requirements kunnen worden opgesteld en worden gemanaged.
- Traceerbaarheid van de functionele en technische aspecten van een informatiesysteem.

Er is software op de markt die deze punten kunnen ondersteunen zoals de requirements tool JAMA en de projectmanagement tool JIRA. Als deze tooling op de juiste manier is ingericht en actief wordt bijgehouden, dan kan dat op de langer termijn een hoop tijd besparen.

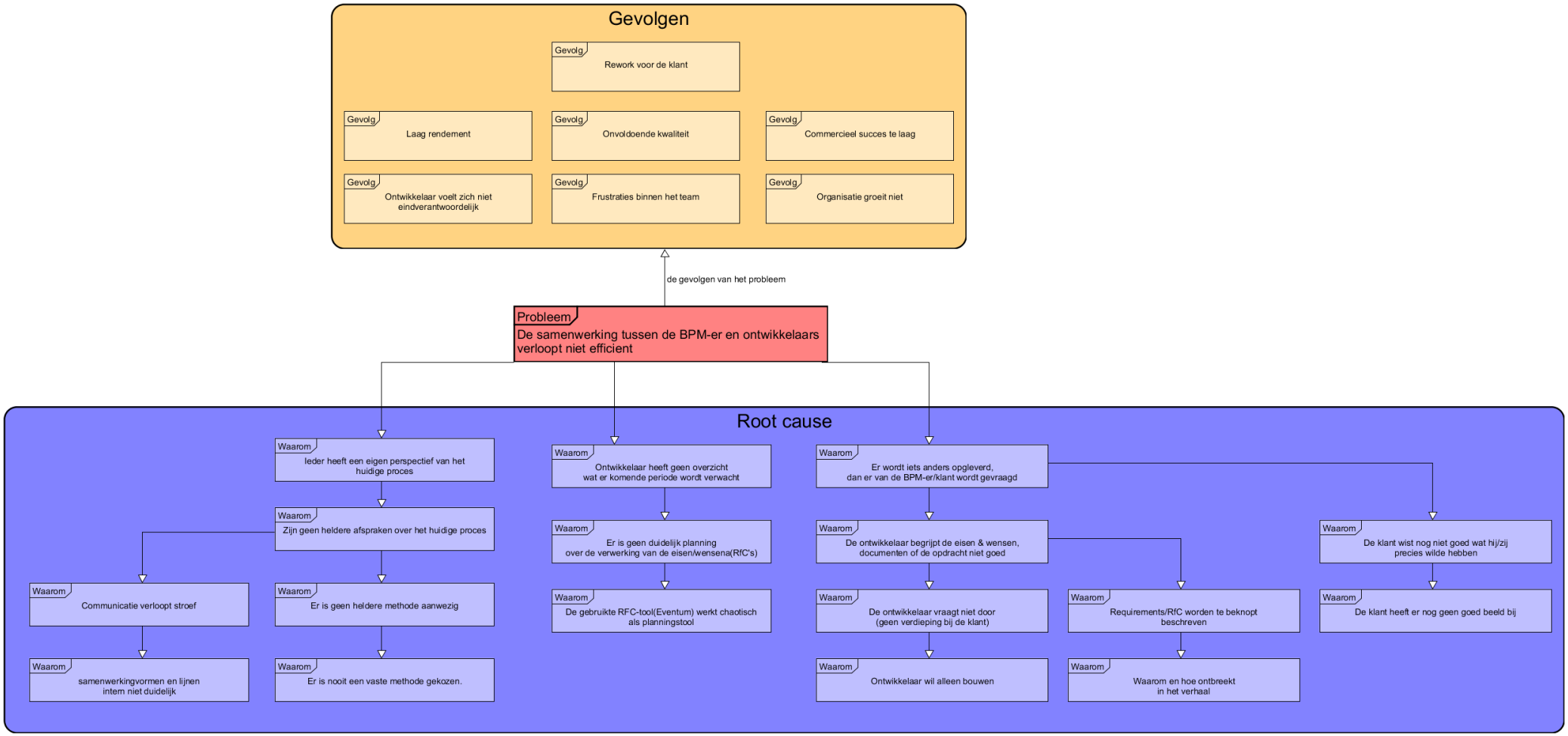
Om de organisatie Builders & Performers in de juiste richting te veranderen, is het belangrijk dat de nieuwe methode met zorg wordt ingevoerd, gemanaged en deels een plek krijgt in de cultuur. In de eerste fase van de verandering wordt ook dit onderwerp meegenomen om de eisen en wensen beter en gestructureerde in kaart te brengen van de klant. Als de invoering van de methode, de business requirements grotendeels heeft gerealiseerd, dan kan er in een latere fase worden gekeken naar het aanschaffen van software of andere hulpmiddelen om de methode van Builders & Performers te ondersteunen.

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

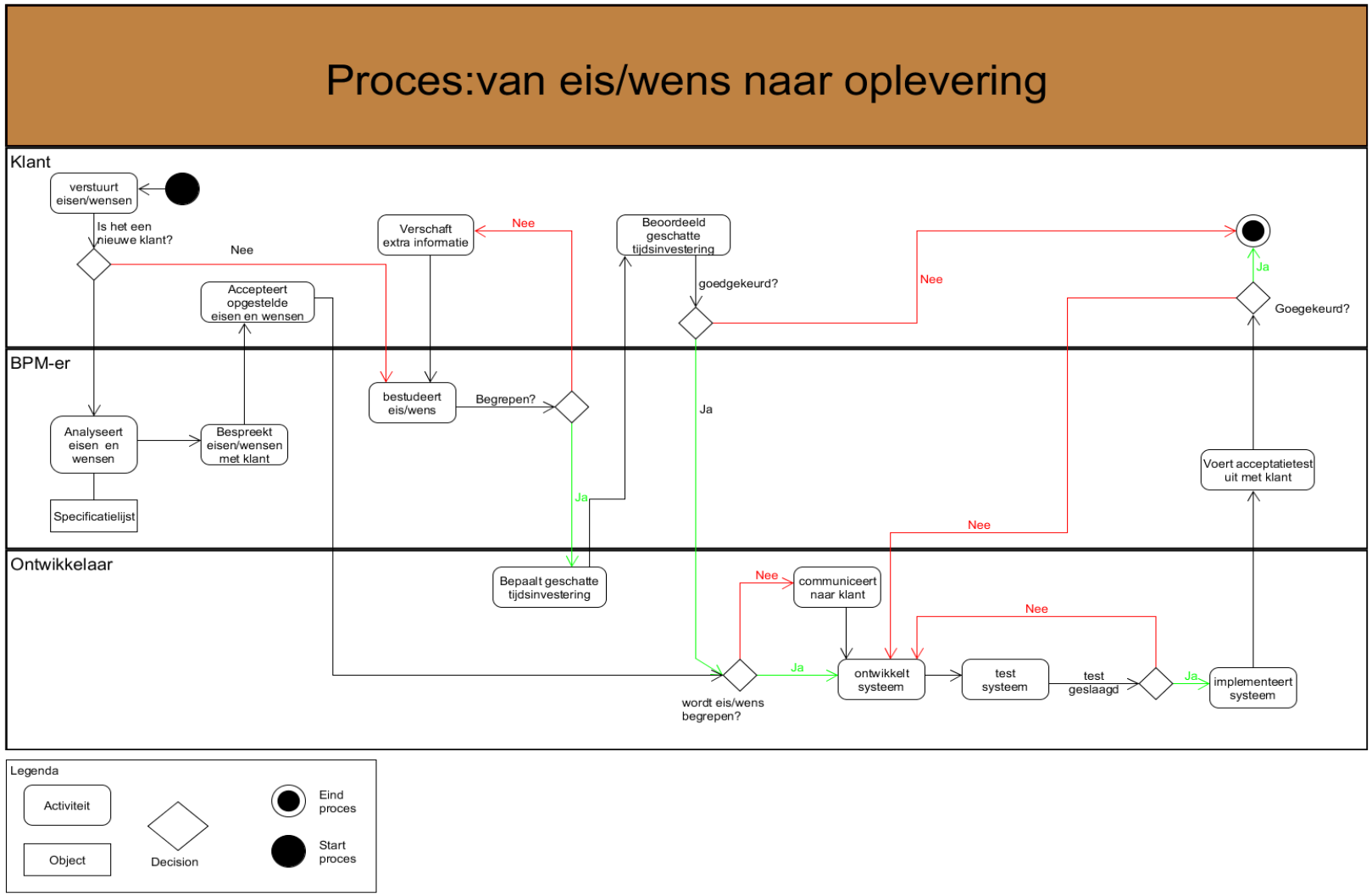
Bijlage A: Huidige en gewenste situatie



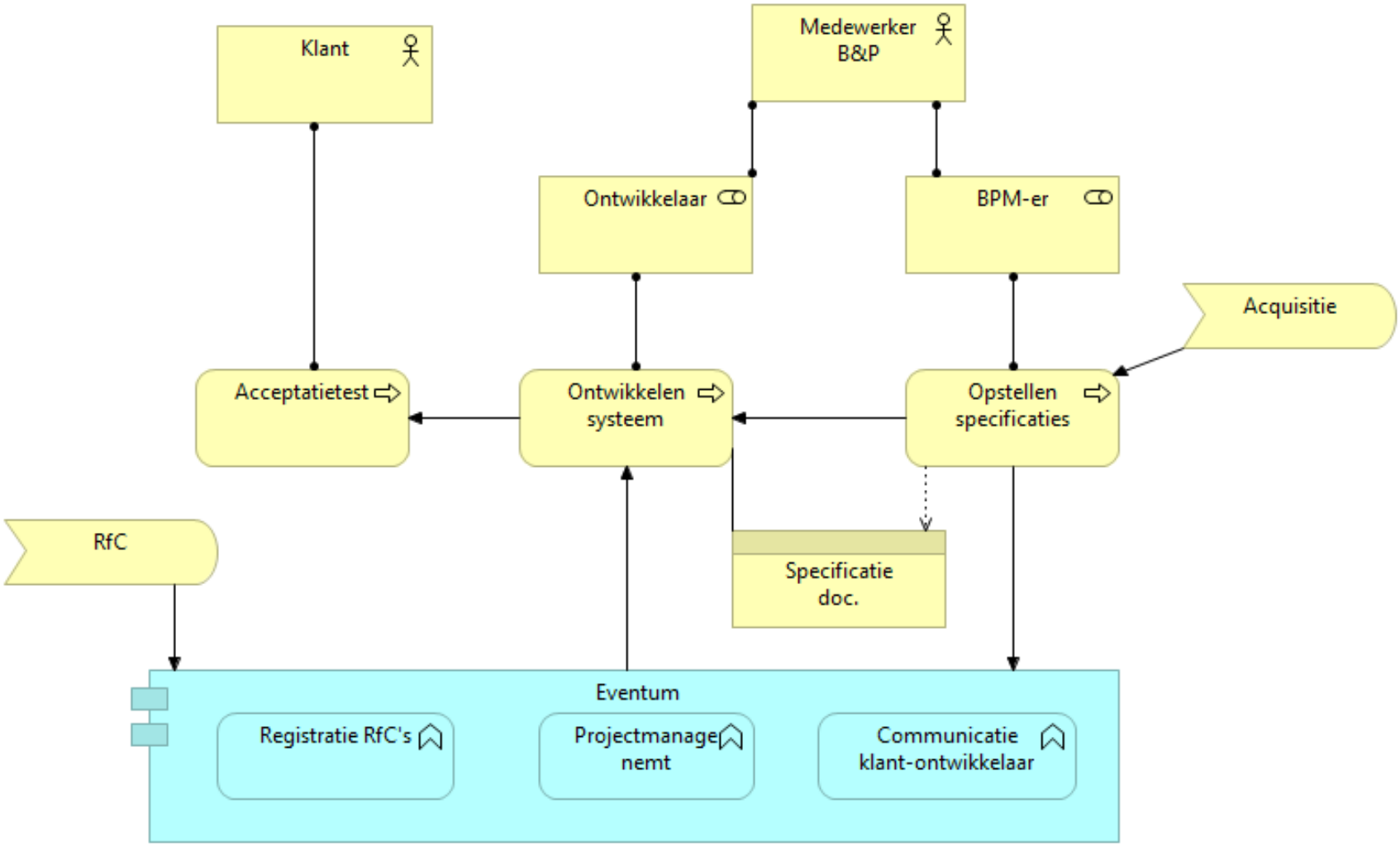
Bijlage B: Rootcause analyse



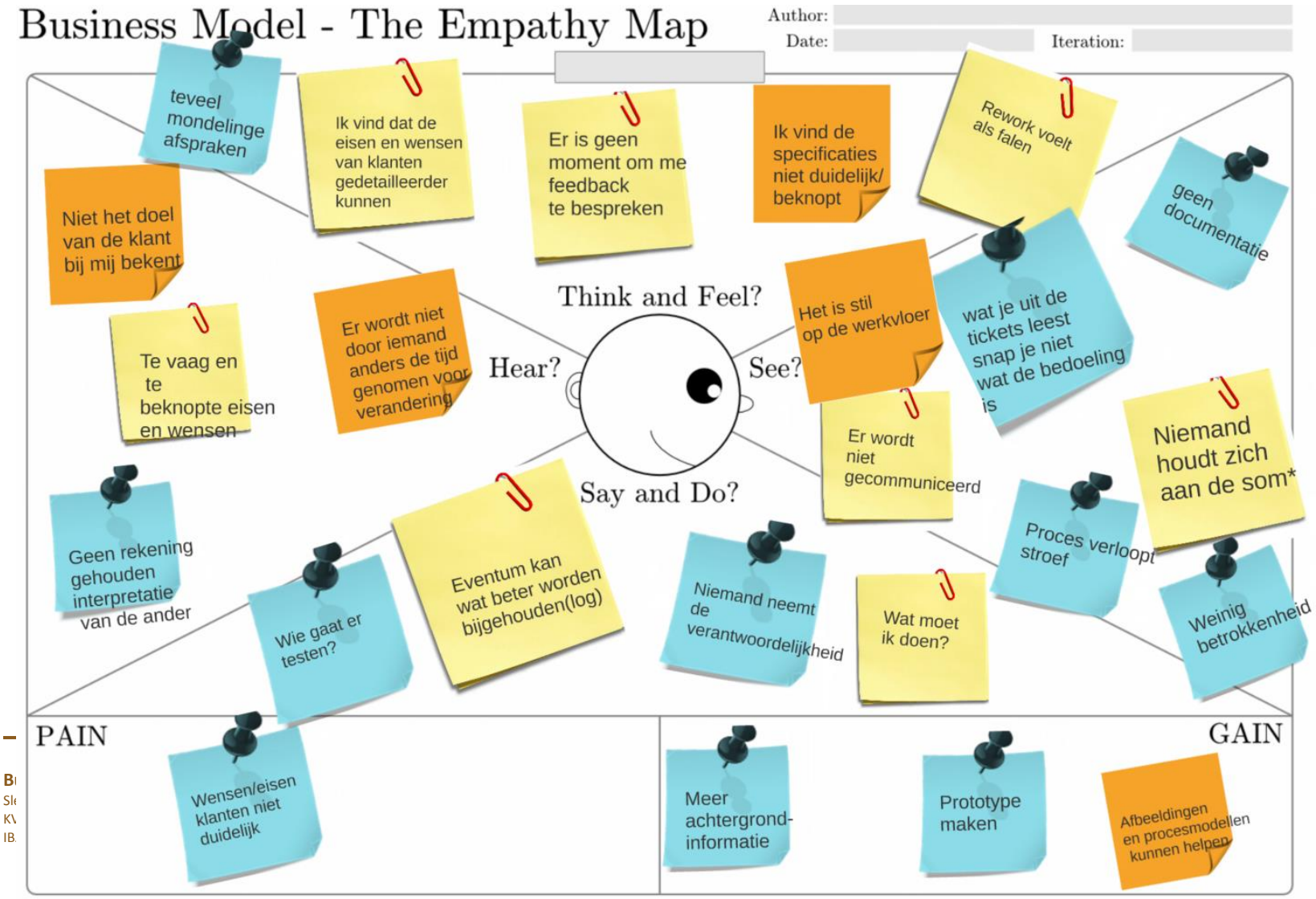
Bijlage C: Huidige proces



Bijlage D: Huidige architectuur B&P



Bijlage E: Customer Empathy Map



Bijlage F: Interviewverslagen

Interview Harry

Hoelang werk je hier al?

Ik ben hier vorig jaar september teruggekomen, en daarvoor heb ik twee/drie jaar ergens anders gewerkt en daarvoor heb ik ook met Stefan, Ronny en Henk gewerkt. Toen was het een andere organisatie. Dus ik ken ze al lang, maar werk hier wat minder lang.

Hoe vind je dat het gehele proces momenteel verloopt?

Haha, jah daar hebben wij al eerder discussies over gehad. Daar zijn zeker wel een aantal pijnpunten uitgekomen waar eigenlijk naar gekeken moet worden. Toen ik hier kwam heb ik ook een voorstel gedaan om hier naar te kijken, maar nooit echt tijd voor gehad want ik ben natuurlijk ontwikkelaar. En om dat dan erbij te toen ..

En wat zijn die pijnpunten dan?

Ik heb voorgesteld om te gaan kijken in de issues van de opdracht. Der is niet echt een duidelijk plan en soms nog echt een vaag idee. Ik heb bij een klant gezeten en daar hebben we een scrum-agile oplossing in combinatie met Jira en daar zitten een aantal methodieken in die je dan heel gedetailleerd met elkaar kan afstemmen, dus daar heb ik wat demo filmpjes van laten zien en daar zouden we ooit is naar kijken. Maar dat is er nooit echt van naar gekomen. Nu zit ik nog steeds met sticky notes en eventum met uitspraken waarvan je eigenlijk niet weet of het nou een opdracht is of iets anders.

Hebben jullie nu een standaard methodiek die jullie gebruiken?

Nou het komt er eigenlijk op neer dat een idee in eventum wordt neergezet en dat moet dan uitgewerkt worden. En die uitwerking gaat nog wel eens mis. Er is in eventum niet echt een goed overzicht van wat uitgewerkt moet worden. En dat er niks uitgewerkt is. Het is gewoon een grote ticket pak, in welke staat het is en hoe duidelijk het is. Vaak details ontbreken. Dat er vaak vanuit gegaan wordt dat je weet waarover gepraat wordt. Dan ontbreken de functionele specificaties van de applicaties. En dan moet je dat even aanpassen. Maar dat is nergens omschreven. En vaak wordt het functioneel wel redelijk omschreven, maar dat het technisch niet goed uitgewerkt is. Dat je het op deze manier gaat aanpassen, dat het of ontzettend in de toekomst heel veel onderhoud gaat worden of andere onderdelen gaat beïnvloeden of andere onderdelen van applicatie zich apart gaan gedragen. En dat daar van te voren niet goed over nagedacht wordt. Op het moment dat je wilt beginnen met implementeren, ineens dat soort vragen moet stellen. Dat is soms niet het beste moment daarvoor. En dan is er ook geen rust voor.

En het is natuurlijk zo als er tussendoor gevraagd wordt, worden mensen geïrriteerd. Dat je er wel echt voor gaat zitten. Zeker als je tien kleine dingen moet doen en op die dingen vragen moet stellen. Jah dan is vaak Henk alweer met een andere opdracht bezig of Stefan in een ander project gedoken zit. Dan moet je continu iemand uit zijn werk gaan halen en dat is niet fijn werken.
Er moet meer focus komen op de specificatie en op de aanpassingen en de technische uitwerking daarvan. Het functioneel goed opschrijven en het technisch goed opschrijven.

De tickets zijn gewoon te beknopt. Dat het gewoon niet goed uitgewerkt wordt, de tekst zelf.

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

Gewoon heel kort wordt neergezet wat de ene denkt, vaak niet met de interpretatie van de ander rekening wordt genomen. Hetzelfde met bugs/meldingen. En dat gaat er mis.
Bijvoorbeeld met invoering van de agenda in de applicatie van een organisatie, dat was een vrij grote aanpassing. En dat het dan niet duidelijk is dat je bijvoorbeeld de agenda gaat wijzigen, dat het jaren terug kan beïnvloeden. Dat agenda's punten worden ingeschoten dat gaat coverteren met de custom agendapunten. En dat je die niet zomaar kan editen. Want dan ga je de rest van het systeem beïnvloeden. Terugkoppeling mechanische bestaat nog niet. Dat soort dingen wordt dan vaak net even gemist. Er wordt te makkelijk over gedacht. Het is ook wel weer fijn dat je die vrijheid hebt. Aan de andere kant is meer duidelijkheid welke kant we op moeten zou wel fijn zijn.

Ik denk ook meer voorbereidend werk , dat bijvoorbeeld iemand een batch heeft opgepakt, en dat met Stefan gaat bespreken, en dan gooit Henk zijn functionele zin neer en dan geeft Stefan een technische zin neer, maar eigenlijk haal je er niet uit wat de bedoeling is als je het gewoon leest. Zeker niet als je niet heel veel voorkennis van de applicatie hebt. Dat is lastig natuurlijk. Daar zie ik wel duidelijk een communicatieprobleem.

De uitwerking van tevoren moet gewoon beter. Duidelijker stappenplannen voor communiceren. Betere schets van wat de gebruiker verwacht. En van tevoren nog een keer gaan zitten wat nou de technische gevolgen zijn, welke keuze we maken. Snel, mooi goed!
Waar vaak niet heel erg bij stilgestaan wordt, dat we hele complexe bedrijfsapplicaties maken. Het is niet allemaal heel recht vooruit. Ieder heeft zijn eigen custom, specifieke dingen, die vaak echt wel over het hoofd worden gezien. En daar eigenlijk geen goed overzicht van is. Als je iemand vragen wat de uitzonderingen zijn bijv op onze Tamcom, dat is eigenlijk een van onze simpele applicaties, dan is er eigenlijk geen overzicht van.

Het enige referentie punt is inloggen op het systeem om te kijken of het nog op de zelfde wijze werkt. En dan moet je maar net op dit ene punt komen waar ze die uitzondering hebben gedefinieerd. En als je er dan gaat langs klikken dan ben je ook de pineut.

Dan komt het er eigenlijk op neer dat je na een technische aanpassing. Henk het volledig functioneel moet doortesten, en daar feedback op moet geven. En dat zou je kunnen voorkomen door overzicht per applicatie te houden wat de uitzonderingen zijn. Op het moment dat het gebouwd werd had iemand moeten vastleggen: voor deze technische functies heb ik een uitzondering gemaakt. Die niet in de standaard zit. En die lijst is ook nooit gemaakt.

Welke problemen kom je nog meer in het proces tegen?
Tooling is op zich wel okee. Jah wat ik wel zou willen hebben is iets meer automatisch testen. Dat zou ik wel fijn vinden. Dat je eigenlijk een flow proces hebt dat je met een druk op de knop naar acceptatie kan, en eventueel naar productie. Dat je daar ook een soort van unitest erbij hebt. Dat je kan zien wat beïnvloed wordt. Dat ontbreekt vaak wel dat je het moeilijk kan achterhalen. Je wilt eigenlijk de functionele requirements kunnen testen. En dat hoeft niet eens automatisch te zijn, maar wel duidelijkheid voor zijn. Maar momenteel bijv bij tamcom geen overzicht is van de functionele requirements. Ik kan alleen maar vragen aan Henk wat vindt deze klant belangrijk. Hij heeft het waarschijnlijk in een documentje staan met afspraken en kan vast een document vastpakken met alle uitzonderingen erin. En dat is er niet.

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

Het is natuurlijk zo als een applicatie groeit en dat het vaak vergeten wordt. Er wordt steeds per project een document opgesteld, maar geen totaal document dat wordt bijgehouden. Het is niet zo in het begin situatie, dit gaan we er aan toevoegen en dan dat eraan toevoegen dan werkt het niet echt heel strikt.

Dat zie je met een andere project ook heel erg, dat je moeilijk kan zien waar de uitzonderingen zitten of dat wordt beïnvloed en dat eigenlijk heel strak in de gaten gehouden moet worden of goed door getest moet worden. Je eigenlijk heel erg afhankelijk bent van een persoon die er eigenlijk door heen klikt om te kijken of alles goed gegaan is. Dat je technisch niet echt goed kan terugkijken of alles goed gaat.

Wordt de eis/wens van de klant verder wel goed begrepen?
Ik heb het idee dat Henk de klant wel goed begrijpt. Alleen de communicatie naar de programmeur vaak al afwijkingen plaats vinden.

Wat voor afwijkingen bedoel je dan?
In technische gevolg trekking. Als je natuurlijk functionele wensen opschrijft zeker als het om wijzigingen gaat. Dan heb je vaak niet het idee welke gevolgen het technisch gaat hebben. Leuk voorbeeld is bij een wijziging van het inschieten van bijlages in de agenda. Dat ze dus van het systeem automatisch een bijlage hebben en volgens handmatig toe te voegen. En dat ze dan niet snappen dat er extra knoppen komen, voor bijlage/dialogen, kan verwijderen, toevoegen, etc. En dat dus zoveel complexiteit met zich meebrengt. Dat wordt vaak niet gerealiseerd. Dat is ook een beetje hoe de klant zijn eigen wensen begrijpt. Daar komt vaak wel extra communicatie over. Dat soort dingen gaan dan vaak wel weer goed.

Dus bij complexere vraagstukken is wel extra communicatie met de klant belangrijk. De communicatie is er wel, is niet altijd even snel , is natuurlijk net hoeveel haast de klant zelf heeft.

Weet je van te voren vaak ook wat je te verwachten staat?
Meestal gaan we van te voren wel zitten, ik heb meestal wel een idee hoeveel werk het zal zijn. Het is alleen met oudere applicaties dat het moeilijk inschatten wordt dat je wel een idee hebt maar niet weet hoeveel extra tijd het gaat kosten om oude dingen goed te krijgen aan de nieuwe situatie.

Omdat daar dus de technische uitzonderingen missen. Dan wordt het moeilijk een detail planning maken, je kan soms stukken code tegenkomen dat je denkt van nou, kost nog wel een dag extra werk. En dat is natuurlijk wel slordig gepland als het nog een halve dag extra moet gebeuren.

Vind je dat het algemene proces wel goed wordt gemanaged?
Op zich is dat wel duidelijk, iedereen weet ongeveer wat die moet doen. Op die communicatiepunten na weet iedereen wel waar die verantwoordelijk voor is. Dan gebeurd dat ook wel, maar de kwaliteit daarvan kan wel iets beter.

Hoe kan de kwaliteit beter?
Door dus een beter gedetailleerder uitwerking te maken. Op zich is het wel duidelijk wie wat moet doen, maar ik denk meer de HOE dat dat beter kan.

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

Je had hiervoor ook bij een ander bedrijf gewerkt, welke methodieken gebruikten ze daar en waarom werkte dat wel of niet beter?
Ik werk al twaalf jaar. Heb bij de koninklijke bibliotheek gezeten. Dat was een klein protyping team. Daar heb ik de digitale bibliotheek van Nederland gebouwd. Dus dat was redelijk direct, dus daar zaten we ook meerdere malen per week met ontwerper aan tafel en technische mensen, dus dat was vrij direct. En daar werden de specificaties ontzettend goed bijgehouden. En daarna heb ik ook een hele tijd gedetacheerd gewerkt bij justitie. En die hebben een gigantische ontwikkeling afgelopen jaren doorgemaakt. Dat je in feiten al standaard in de productieomgeving zat te scripten, met de klant aan de telefoon tot aan hele gespecialiseerde overheid systemen.

We hadden in die organisatie ook een heel uitgewerkt scrumproces. En dat werkte ook. En je werkte per week ook aan zes verschillende producten en in een maand wel viertig. En dat soms in vijf verschillende talen. En systemen bij justitie draaien soms wel tien jaar. Nou dan kreeg je code van tien jaar terug te zien en dan ga je er scrum op toepassen en dat dan de documentatie niet up to date is, dan ga je dat meenemen. Dus dan ga je proberen jouw aanpassingen goed te documenteren en daar zat ook wel veel uitdaging in. Maar dat heb ik wel goed onder de knie gekregen.

Maar daar ging het wel echt te ver. Te veel meeting, te veel overhead, te veel klanten. Volgens mij was ik per week maar 2,5 dag aan het programmeren, de rest alleen aan het ouwe hoeren. Installatieaccounts aan het regelen, handleidingen schrijven, mensen aan het rondbellen en emaillen. Daar zit ik hier niet op te wachten. Daar was natuurlijk ook het budget voor. Maar dan raak je de complete dynamiek kwijt commercieel gezien. Complete dooddoener zijn. Dat zijn twee hele extreme zou maar zeggen.

Maar ik denk de methodieken die daar inzaten. Ik heb dus de hele invoering van scrum enzo meegemaakt en de kaart technieken die daar bijzaten. Tooling die ze daarvoor gebruikte en dat daar wel heel veel goeie dingen inzitten. Ik denk dat we daar best wel hele goeie dingen uit kunnen pakken om de boel te verbeteren.

Maar je moet niet gaan zeggen we gaan deze methodiek toepassen, want dan heb je echt iedereen zeg maar tegen je. Maar dat je dus moet kijken, er zijn methodes, zoals agile enzo, die bieden heel veel vrijheid en zitten heel veel truckjes in omdat te verbeteren.

Zoals bijvoorbeeld inschatten hoeveel werk het is. Daar hebben ze dan zon scrum pokergame voor. Door zo'n soort punten systeem te hanteren, denk ik dat het veel realistischer wordt. Nu is het vaak zo dat Stefan roept een half uurtje, en dat wij geen idee heeft waar die het over heeft. En dat we dan moeten zoeken na een halve dag waar we moeten zijn. Daar krijg je dan hele rare tijds schattingen voor.

Ik denk dus dat het belangrijk is om daar met elkaar een afstemming op te krijgen dat we van elkaar ook weten wat de ander wel en niet complex vindt. Misschien tijdens het inschatten van tijd elkaar al punten kan geven waarom hier hoog of laag wordt aangegeven. Dat vind ik ook al een hele interessante ontwikkeling.

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

En het overzicht. Wie is waar mee bezig, dat is behoorlijk lastig. Het is momenteel een aantal gemarkeerde bugs in een lijst en eigenlijk de status daarvan dat je ze een voor een moet openklikken. Het is vrij moeilijk om te kijken of er al een idee is, of ik er al iets mee kan, waar kan ik een stukje bij gaan schrijven, dat is vrij moeilijk. Als iemand een idee had erop dat het vaak in een hoekje wordt gezet en dat iemand het gaat uitwerken is niet heel groot. Ik heb wel gemerkt dat JIRA daar hele goeie overzichten voor heeft.

JIRA werkt dan met sprints. En dat doen we een soort van ookwel. In het begin gaan we kijken welke tickets we mee moeten nemen in deze batch. Maar bij een agile proces hoor je achteraf ook nog even te gaan zitten om te vragen:

- Wat ging er deze keer goed
- Wat kan er beter?

En dat doen we eigenlijk niet. We schelden wel eens op elkaar. Soms wel eens tijdens de lunch. Maar om nou echt te zeggen dat we ergens in de week een momentje hebben is er eigenlijk niet. Eigenlijk is het bijna verplicht om een uurtje per week te zitten van wat kan beter , en dan niet alleen maar uit te spreken, maar dat op te schrijven en daarvoor een reminder te maken. En dan ook de volgende keer kan zeggen wat kan beter en wat is er met de punten van volgende week gebeurd. Die feedback loopt werkt bij ons niet heel erg goed.

Zoals ik al zei, toen ik hier kwam dat we hier echt een keer naar moeten kijken maar daar is eigenlijk nooit tijd voor. Maar het idee is wel om hier naar te gaan kijken. Ik heb dan ook wel eens filmpjes laten zien hoe dat dan moet gaan werken en daar was dan ook zeker wel interesse naar. Maar om de tijd te nemen om JIRA hier in te richten en te laten zien hoe dat werkt dat is er eigenlijk nooit van gekomen. En het staat nog ergens op me to-do listje. Maar om EASE naar een beter niveau te krijgen was wel even belangrijk.

Het is lastig over te stappen bij een bestaande klant. Het is makkelijker om het te testen bij een nieuwe project.

Hoe zou jij het graag anders willen zien?
Ik denk dat ik de meeste dingen wel gehad heb. Het is inderdaad iets meer overzicht in de actievere werkzaamheden. Wat dat betreft is klantenregistratie voor klanten issues echt perfect. Voor de actieve development of de technische kant, wat eigenlijk niet voor de klant belangrijk is maar voor ons, doet het echt onder aan hoe het moet functioneren. Het is meer soort van eerste lijns ticketsysteem. Maar daarnaast meer development traject met meerdere onderdelen zijn. Dat mis ik ook. Je kan wel een heel ticket hebben die zo groot is met honderdduizend dingen. Daar zijn eigenlijk duizend taken in te definiëren. Het is gewoon zo groot en ga maar rennen, maja in JIRA heb je dat je het dan kan opsplitsen in mooie taakjes.

En dat kan nu niet, waardoor nu de ene heel veel werk heeft en de ander een heel stuk minder. Ik denk dat dat wel echt een van de belangrijkere dingen is. Ik denk eigenlijk als je de JIRA tickets zou gaan uitwikkelen, dat het automatisch met het proces mee gaat komen.

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

Interview Stefan

Hoelang werk jij voor B&P?
Sinds het begin, dus dat is in 2013 toen het bedrijf werd opgericht. Toen zijn we met zijn vijven begonnen.

Hebben jullie in het begin bepaalde processen gedefinieerd hoe jullie te werk gaan?
Nou de eerste paar maanden was een soort van overleven. Maar na het overleven moest het gestructureerd worden.

Hoe zie je het proces nu vanaf het moment dat een eis/wens van de klant komt tot aan de oplevering?

Hoe ik het proces nu zie is dat de klant eerst tegen de BPM-er aan babbelt, en die vist de eerste mogelijke en niet mogelijkheden er wel uit. En dan wordt het met mij of Ronny besproken. En dan wordt er concreet besproken wat we gaan doen.

En hoe vind je dat proces nu verlopen?
Af en toe nog wel stroef.

Wat kan er dan beter/gaat er fout?
Soms heeft Henk al bepaalde oplossingen voor ogen of heeft een bepaald idee hoe dingen werken? En daardoor is er een soort spraakverwarring. Omdat bepaalde dingen heel anders werken.

Hoe vind je de lijst met eisen/wensen aangeleverd door de BPM-er?
Ik vind het wel erg beknopt en soms zelfs te beknopt. In mijn optiek.

Wat zou hier aan verbeterd kunnen worden?
Wat wel zou helpen is wat meer achtergrond informatie, hoeft natuurlijk geen compleet boekwerk te worden. Maar met wat meer informatie kom je tegen beter in de goeie oplosrichtingskant. Globaal procesbeschrijving is altijd wel handig, hoeft natuurlijk ook niet tot in de mega details, want daar heeft de programmeur ook niks aan. Het is natuurlijk wel handig dat je het in context kan plaatsen.

Zijn er daarnaast nog andere problemen die je in het proces tegen komt?
Nee dat valt op zich wel mee, **het is echt puur de kennisoverdracht van wat de klant wil en dat bij ontwikkelaars laten landen**. Daar zit wel het grootste punt. Kijk, als we eenmaal in het ontwikkelproces zitten hebben we daar wel een strak proces voor hoe dat moet. En als iedereen zich aan houdt inclusief Henk, dan gaat dat wel goed.

Welke methodieken gebruiken jullie in de organisatie?
Nee, wat we doen is een mengelmoesje van methodieken. Maar we hebben ze nog nooit benoemt of opgeschreven. We hebben een aantal dingen wat we standaard doen, maar dat is nooit echt opgeschreven.

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

Wordt die " standaard" ook gehanteerd?
Er zijn wel bepaalde processen of oplossingen wat we moeten registreren en het bijhouden. En het ontwikkelgedeelte is kort opgeschreven en dat weten de ontwikkelaars ook wel.
En dat gaat meestal wel goed, als men dat doet. Maar als niemand zich eraan houdt die methodieken, dan heb je er ook niks aan.

Hoe vind je de tooling die in het proces gebruikt wordt?(en dan met name eventum).
Jah eventum is eigenlijk net een soort excel sheet, het belangrijkste vind ik dat er gewoon een goed registratiesysteem is van issues. En dat doet eventum. Daar zijn altijd dingen die wel anders kunnen, maar dat heb je bij andere tools ookwel. Het in ieder geval handiger dan excel. Eventum is wel handig omdat we het aan onze sourcecode reposterie is geïntegreerd. Dus bij elke wijzing aan source code weet je aan welke issue het van eventum het is gekoppeld. Dat vind ik het handige van Eventum. Dat je kan zien dat de code is gewijzigd en waarom het is gewijzigd.

Hoe zie je het zelf graag anders?
Jah is naar mijn gevoel af en toe hakken op de tak. Je krijgt dan functionaliteiten of steekwoorden van wat er gemaakt moet worden. Dus dan heb ik een bepaalde interpretatie daarvan. Dus ik moet Henk helemaal uithoren of het weer terug moet. Dus jah, iets meer op papier is wel handig. Kijk, het wat staat er meestal wel, maar de waarom mist.

Weet jullie van elkaar wat jullie verantwoordelijkheden/rollen zijn?
Ik weet wel van iedereen wat ze primair horen te doen. Soms stap je wel eens buiten je verantwoordelijkheden, wat niet erg is. Nee ik denk dat dat op zich wel goed gaat.

En wat hoor je in de omgeving van het proces?
Ik hoor van andere ontwikkelaars dat het vaak ook te vaag is of te beknopt is. Aan de andere kant, ik ben van mening dat een programmeur niet echt precies alles hoeft te snappen. Maar in ieder geval genoeg dat die het kan maken. De regels weten van een proces kan bijvoorbeeld wel handig zijn.

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

Interview Menno

Hoelang werk je hier?
Ongeveer vier maanden

Dat is nog een korte periode, hoe is je eerste indruk bij b&p?
Procesmatig was het nog wel even wennen. Altijd lastig als je pas moet beginnen, maar er is verder ook geen documentatie. Er zijn verschillende applicaties en dat is lastig. Ik ben begonnen met hele simpele opdrachtjes. Wat opdrachten die verzonnen zijn voor mij om het systeem te leren kennen. En vanuit die basis verder gestroomd, kleine bugs of fixes gemaakt, dus wensen van de klant. Stefan selecteert een paar leuke opdrachten voor me die ik kan doen. Om dan even kennis te maken met de applicaties van de klanten. En op die manier rustig daarin gerold. Met Eventum moeten wij deze dingen geprogrammeerd hebben, invoeren of wel inchecken noemen wij dat dan. Op die basis kunnen we antwoord geven aan de klant of er vragen onduidelijk zijn of als er extra dingen gevraagd worden die oorspronkelijk niet in de echt specificaties stonden. Die zou ik dan terug naar Henk moeten terugschakelen van klopt dit dat we met de klant hebben afgesproken. En soms kan je dat als ontwikkelaar zelf doen.

Hoe vind je dat proces nu lopen, dus van het moment dat de eisen en wensen van de klant binnen komt tot aan de oplevering?
Ik denk dat het wel iets duidelijker kan.

Wat kan er dan duidelijker?
Duidelijker in de zin van wat er uiteindelijk echt afgesproken wordt, bekend is van die issue in eventum. Eigenlijk zou de specificatie gelijk duidelijk moeten zijn. Jah eigenlijk dat wel. Iets gedetailleerder mag wel. Het hoeven geen hele documenten te worden. Het zou misschien helpen om vanuit de context van de klant kunnen zeggen dat ze willen bereiken met de applicatie, want ze hebben er een bepaald doel mee. En ze willen een bepaalde functionaliteit toevoegen en in welke context dat dan geplaatst moet worden en dan kijken wat nou het echte doel van de gebruiker is. Misschien is het handiger om met wat plaatjes of simpele stappen precies laten zien wat ze precies willen. Procesoverzicht met wat stappen erin geplaatst. Wat plaatjes om het overzicht completer te maken.

Werken jullie hier ook met een softwareontwikkelings proces?
Nee.

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

Welke problemen kom je nog meer tegen in het proces ?
Nee niet echt specifiek. Ik kom wel problemen tegen, maar dat is vaak technisch of het is niet duidelijk van aard wat er afgesproken is. Teveel mondelinge afspraken of je kan het nergens terugzoeken en dat soort zaken. En de verantwoordelijkheid moet duidelijk zijn. Als je wat moet opzoeken, zeg het dan van okee je gaat er vanuit dat je dit vanzelf allemaal opzoekt. En dat is niet altijd even duidelijk.

Hoe vind je de tooling zelf werken?
Op zich werkt het wel okee, er mogen wel wat initiators toegevoegd worden. Zoals als er handmatig werk gedaan moet worden, aparte velden bij zijn, maar dat kan Stefan aanmaken. Of dat we zeggen, let op, we zitten nu in de sharp time, manual action is dat dan als je voor de klant een acc of productie aangepast moet worden. Dat soort kleine zaken weet je wel. Jah misschien nog wat andere statussen. Maar daar zou ik nog over na moeten denken. Voor de dingen die we nu doen werkt het wel goed op zich.

Voor jouw ook een duidelijk overzicht wat je moet doen?
Jah als het duidelijk wordt neergezet dan wel jah. Ik kom niet gauw voor verrassingen te staan.

Hoe zou jij het zelf graag anders willen zien?
Voor mij zijn er ook vaak punten onduidelijk, dat betekent dat ik er heel erg op moet gaan hameren. Als ik bezig ben, en werk er ook pas net, kom ik vaak wel dingen die heel erg onduidelijk zijn. Technisch gezien dan. Daar moet ik altijd op Stefan voor op terug vallen of eventueel Ronny. Misschien is het handig dat er ook wat meer dingen in eventum worden ingevuld, van ik loop hier en hier tegen aan. Dat kan nu ook al met internal notes, misschien handig een soort log waarop je stuk gelopen bent en dat een ander kan teruglezen dat waarop is vastgelopen hoe het is opgelost, misschien is dat wel handig.

Wat hoor je van andere mensen hoe hier wordt gewerkt?
Nou het is soms een beetje “schelden” voor wat voor poep er achter is gelaten op het systeem. En dat hoor je heel vaak van Harrie natuurlijk. Jah zelf zie ik dat ook wel.

Zijn er volgens jouw ook nog winpunten?
Als iemand druk is, of niemand reageert op elkaar. En je loopt ergens op vast dan probeer je dat nog wel een keer aan te kaarten. Inderdaad wat ik zei als dat nog gelogd kan worden. Dat mensen dan op een bepaald tijdstip ernaar kunnen kijken, maar dat dat dan ook een keer besproken wordt. Op het gebied van communicatie kan het dus nog wel een stuk beter.

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

Interview Henk

Onderwerp: Probleemstelling en oorzaken
Datum: 16-02-16

Wat speelt er bij builders en performers?

Uiteindelijk zijn we heel kritisch, want hetgeen dat de ontwikkelaars uiteindelijk bouwen bepaalt het succes van de organisatie**. Dus als zij iets heel snel kunnen bouwen, dan zijn wij goedkoper, kunnen wij commercieel succesvol zijn en kunnen wij meer werk verrichten waardoor we ook de huidige klanten ook kunnen blijven bedienen.** Dat je gewoon capaciteit overhoudt op deze manier.
Wat ikzelf merk, en dat is eigenlijk al twee jaar aan de gang dit moment binnen de organisatie, is dat: ik spreek met klanten, ik doe vaak een inventarisatie van de wensen, de ene keer vertaal ikzelf de wensen naar functionaliteiten en geef ik het door aan de ontwikkelaars, de andere keer geef ik een samenvatting door. Vraag ik of de ontwikkelaar de wensen zelf vertaald naar functionaliteit.
Het proces, zodra de opdracht is aangenomen, er precies helder is wat ze moeten maken. En als ze eenmaal gaan bouwen, komen ze terug met vragen over het resultaat. Daaruit blijkt dat de ontwikkelaars zich eigenlijk niet hebben verdiept in de functionele wensen . Ze hebben heel snel perceptie van ik snap het en heb een oplossing, ik kan er een tijdsbesteding aan vastzetten, en dan gaan ze aan de gang, zonder in werkelijkheid echt te begrijpen wat de wensen zijn. En dat is op basis van resultaat.

En daarnaast valt me op dat de ontwikkelaar blij is als die ze werk goed verricht heeft, en zich niet bekommert over het eindresultaat. En jah, als je werk voor 80% doet, dan weet je zeker dat de klant terugkomt voor de overige 20%. En dat is iets wat je wilt voorkomen, want de interactie met de klanten en foutherstel en de totale tijd hieraan. En dat kost heel veel, zeker in vergelijking met het project, of de oorspronkelijk opdracht zelf. En dat proces vind ik op dit moment echt te kostbaar. Uiteindelijk komen wij er wel, maar met teveel iteraties en kost te veel tijd.

En ik vind de kwaliteit niet vaak goed genoeg, dus we hebben altijd puur het minimale eruit gehaald wat we zouden moeten doen, maar heb het idee dat we veel betere software kunnen opleveren.

Probeer me even te beperken tot het probleem, oorzaak benoemend:
Heb ik het idee dat het in de aard van het beestje zit dat de ontwikkelaar alleen wil gaan maken. Dus het gesprek dat die voert met de BPM-er of soms de klant doet die maar omdat het moet. Maar hij voelt zich veel fijner als die sneller achter de computer kan zitten en al dingen maken wat er van hem gevraagd wordt. Ik heb het idee dat dat echt het aard van het beestje is. En het lukt ons heel slecht om daardoor heen te breken.

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

Het is dus vaker voorkomen, waarom gaat het dan nog steeds fout?
Het vakmanschap wat ze geleerd hebben, waarom ze voor it hebben gekozen en graag de techniek willen leren kennen. En zodra ze achter hun computer zitten en de code kunnen kijken begrijpen ze de wereld. Het geen wat vaak de klant wil, daar zit vaak een probleem of behoefte achter, en dat ze daar weinig affiniteit mee hebben. En daardoor kunnen ze zich niet inleven:

- Waarom bepaalden rapporten worden gevraagd
- Waarom een oplossing niet als klantvriendelijk wordt ervaren

En zodra je praat, moet er echt een dialoog zijn, ik zeg wat, dan moet je begrijpen wat een ander zegt en anders moet je doorvragen. Als het nou de BPM-er is of de klant is, maakt niet zoveel uit, je moet uiteindelijk werk aannemen.

Maar is dat dan ook niet meer de taak van een BPM om de eisen en wensen naar functionaliteit om te zetten?
Ik denk dat het niet uitmaakt of je met de BPM-er spreekt of dat je met de klant spreekt. Laat ik het zo zeggen. In de praktijk, als ik heb gesproken met een accountantskantoor en ik kom daar samen met de klant tot een oplossing, om ook zijn klanten weer van een oplossing te voorzien. Dan laat ik vervolgens een offerte zien, zien wat de rollen zijn, een stukje functionaliteit en naar mijn idee is de oplossing al beschreven de rest is details invullen. Ook mijn opdracht moet worden aangenomen, en uiteindelijk moet er worden doorgevraagd, ik heb iets beschreven, maar wat ik schrijf interpreteert de ander weer vaak als iets heel anders. Dus ik denk dat je bij de opdracht aannemen, reten goed moet begrijpen wat je wilt gaan bouwen, voordat ik kan zeggen dat het ook daadwerkelijk kan. En daar zit een gat eigenlijk in de afgelopen twee jaar.

Kan je een aantal voorbeelden noemen?
We hebben een management dashboard gemaakt voor een organisatie. Daarin heeft de klant een eigen BPM-er aangewezen, die heeft bepaalt welke informatie er moet komen in het management dashboard. Die heeft daar een onderverdeling in gemaakt en heel gedetailleerd functioneel ontwerp gemaakt . Uiteindelijk is daar een sessie voor georganiseerd. Dat de programmeur de klant een op een heeft gesproken. Je zou denken dan, er bestaat al een FO, er bestaat al een doel en randvoorwaarden waaraan het moet voldoen. Op basis daarvan wordt een PvA gemaakt en heel concreet de informatie wordt verzameld, zodat de uitvoerders zijn werk kunnen doen. En dan wordt alsnog vaak niet begrepen wat de klant bedoelde, de eisen en wensen stonden dus op papier, maar als je ging doorvragen begrepen ze eigenlijk helemaal niet wat er op papier stond of wat we uiteindelijk moesten opleveren.

Je hebt een proces, daarin kan je uiteindelijk een functioneel ontwerp maken, maar uiteindelijk ga je vaak mondeling ga je toetsen in datgene dat jij gelezen hebt of jij het begrijpt ook correct is. Het is een heel moeilijk proces, iemand neemt werk aan en los van dat de ander al het voorwerk doet, betekent dat er toch een soort overeenstemming moet komen wat het daadwerkelijk moet worden. Dat zowel de klant het begrijpt als de programmeur.

Wat de klant verwacht, ik heb het idee als ik het plan met de klant beschrijf, dat daar vaak heel snel een overeenstemming wordt bereikt met wat het gaat worden.

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

Ander voorbeeld wat goed is uitgewerkt
Agenda delen hebben wij gerealiseerd, daarbij heeft de klant zijn wens uitgesproken, die heeft ook bij de concurrent een offerte aangevraagd. Uiteindelijk heb ik met Ronny een plan uitgewerkt, daarin zijn allemaal plaatje gemaakt met procestekeningen en daar is een heel document aan randvoorwaarden opgesteld. En heel concreet hebben wij gezegd, dat het idee dat de medewerker de gehele agenda kan bewerken en indelen, en zoveel mogelijk met de hand kunnen aanpassen. Uiteindelijk gaan we aan de gang, een programmeur doet de analyse van de requirements , creëert zijn eigen plan van aanpak, gaat ermee aan de gang en dan zie je dat eigenlijk alles wat daarvoor is gemaakt wordt vervaagd, en eigen plan van aanpak wordt leidend.
Er wordt dus eigenlijk nog watervalachtig te werk gegaan, en dus niet naar teruggekeken wat de daadwerkelijke requirements waren?

Daar lijkt het inderdaad vaak wel op, zelf kan ik ze vaak ook wel weer aanwijzen en dan blijkt uit eens, ooh jah, maar zo heb ik het nooit gesnapt of jah dat is een bizarre requirement, die kan je zo echt niet opstellen. Terwijl dat eigenlijk met de opdracht aannemen al besproken moet zijn.

Dus met andere woorden, de koppeling mist?
Jah, maar we steken er absoluut wel tijd in om die overdracht te verzorgen en toch is het resultaat niet goed genoeg.

Wat hoopt er uiteindelijk bereikt te worden?
Uiteindelijk zoekt de organisatie een werkmethode , in een ideale wereld, doet een BPM-er een vooronderzoek, die komt tot een globale opzet van een oplossing en een globaal ontwerp. En vervolgens heb je de uitvoerders die het op de werkplek zelf realiseren. Dus een BPM-er doet een analyse van de behoefte, die maakt concreet een oplossing situatie schets die(conceptuele oplossing). De rollen worden beschreven, er wordt beschreven de inhoud die je van de organisatie wilt bijhouden. **Ik ga ervan uit dan als dat in grote lijnen afgesproken is met de klant, dat in mijn ideale wereld een programmeur in staat moet zijn om met het geleverde materiaal en de afspraken die zijn gemaakt een gesprek moet voeren en het vervolgens kan uitvoeren. Dat is wat ik verwacht.**

Dus je hebt een algemeen verhaal, een algemene schets van wat het gaat worden/wat het gaat doen, dus eigenlijk heb je de deelresultaten geformuleerd. En uiteindelijk moet de programmeur dit in blokjes opdelen en het per blokje het gaan uitvoeren. Dat is de ideale wereld, dat is vaak ook het goedkoopst voor de organisatie. Dus de verhouding van wat een programmeur zou moeten doen en wat een BPM-er zou moeten doen. Dus de opdrachtgever, de klant, de eindopdrachtgever wordt van de programmeur. En dat de BPM-er ter ondersteuning is voor onderzoek en informatie input.

Zijn er andere factoren die nog een rol kunnen spelen in het proces(denk daarbij aan tooling/methodieken)
We hebben het vaak geprobeerd, maar ik denk dat de samenwerkingsvormen en lijnen intern niet duidelijk zijn, en dat dat eindelijk ervoor zorgt dat de mensen niet weten wat hun eindverantwoordelijkheden zijn.

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

Dus er is nog geen structuur/methodiek in de organisatie?
Ik denk als je straks iedereen een op een gaat interviewen dat ieder een apart beeld heeft van eigen methodieken en structuur.

Ik ga jouw een aantal voorbeelden sturen die ik ooit ontworpen heb hoe een organisatie zou moeten werken, maar als je naar de uitvoerders gaat kijken zie je het niet .

Waarom?
Ik denk dat er vanuit het ontwikkelgebied te weinig verantwoordelijkheid voelen om het te handhaven en degene die het heel erg zouden willen te afgeleid zijn. Ik denk dat dat ook al een onderliggende oorzaak kan zijn van het probleem.

Het probleem speelt zowel bij nieuwe klanten als bij bestaande klanten(als ze bijvoorbeeld een RFC inschieten)

Mijn ervaring is ook als wij een klant hebben die het heel goed heeft georganiseerd, eigen projectleider heeft en precies bij ons heel duidelijk communiceert wat er moet worden opgeleverd en wat de inhoud moet zijn. Dat we dat een stuk makkelijker vinden, en zodra we een klant hebben die het aan ons overlaat zie je dat het vaak chaos wordt.

Is het niet effectiever om alles bij het ouden te laten?
Ik denk als we blijven werken zoals we nu doen, dat we nooit kunnen groeien. Dat is onmogelijk. En daarnaast vind ik ons commercieel succes te laag, het rendement dat we draaien met de huidige productie die we draaien te laag.

Ik hoop dat er met een nieuwe methode dat het werk makkelijker wordt uitgevoerd, zonder dat ik er niet teveel tijd in hoef te besteden. Dus wel interventarisatie doe, en dat de overdracht van de eisen/wensen redelijk zelfstandig kan worden uitgevoerd. Met minder fouten.

Uiteindelijk heb je gewoon een projectorganisatie zitten, en ik denk dat de uitvoerende mensen in staat moeten zijn de opdrachten die ze krijgen gewoon uit te voeren zonder continue sturing van buiten af. Heb je het wel getest, heb je het wel opgeleverd.

Het gaat procesmatig niet heel strak. Ik heb het idee dat er eerst heel weinig begrepen wordt van wat zou het moeten doen en wat de eisen/wensen betekenen. Er komt altijd een stukje creatief proces, want datgene wij maken bestaat nog niet , daar zit een creatief proces aan. Om een creatief proces te realiseren moet je echt wel, voordat je een opdracht aanneemt goed doorvragen wat precies de klant bedoelt. Het creatieve proces mist in mijn beleving klantbehoefte en de uitvoering, ik denk dat de rolverdeling niet helemaal duidelijk is. Waardoor daar fouten in worden gemaakt. En in de procedure tot oplevering dat het ook niet helemaal strak is. En dat heeft allemaal effect op het eindresultaat.

Ik denk dat we organisatorisch echt een probleem hebben, en niet een klein beetje ook. We zijn nu met drie ontwikkelaars, ik denk als het team verdubbeld dat Ronny in een minuut overspannen is, dat ik overspannen ben .

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

En dat moeten we niet goed zetten. We zijn nu in gesprek met klanten waarvoor grote projecten kunnen worden aangenomen, als dat doorgaat moeten we ineens nog meer mensen aansturen en het team aansturen en dat moet niet op deze manier. Natuurlijk is het heel makkelijk om een teamleider op te stellen, die het werk verdeeld en een extra business proces manager aan te stellen. Die nog meer uitwerkt, maar jah, dan gaat je kostprijs zo omhoog dat je niet meer betaalbaar wordt.

Het probleem speelt al een tijd in de organisatie, hebben jullie hiervoor zelf ook al oplossingen voor bedacht?
Ik heb eigenlijk meer een soort scrummethodiek, iets meer dan een half jaar geleden geïntroduceerd, hoe werk je als project, wat verwacht je van elkaar, bij wie moet je opleveren. Dat hebben we toen besproken, hebben we zelf vooruitgang geboekt op die manier. In die zin hebben wij het aannemen van de opdracht, dus het bepalen wat het moet worden, dat daar dus een gat lag, en dat daar veel meer aandacht aan besteden moet worden. Dat het beter is om meer tijd te steken in de analyse van de opdracht, dan al gelijk aan het bouwen. Dus dat is al de eerste vooruitgang die we hebben geboekt. Je zag dat de ontwikkelaars daar echt moeite mee hadden.

Ook een probleem is qua planning, we hebben het systeem Eventum, waar alle RFC's in worden verwerkt. Maar het is geen planningstool. Het is nu vaak dat in het begin van de dag Ronny vaak werk verdeeld, dan denk ik een organisatie in 2016 het redelijk ouderwets is dat een baas tegen je zegt, jij moet dat doen en jij moet dat doen, etc. Daarom heen loop ikzelf ook nog vaak te klagen als iets niet op tijd is of niet goed wordt geleverd. Dus degene die het hardst schreeuwt, krijgt daarin ook voorrang.

Ik denk in de chaos hoe ik het nu beschrijf, dat het wel invloed heeft op het functioneren van de ontwikkelaar. Die voelt altijd een druk dat er nog meer werk aankomt.

Per dag krijgt een ontwikkelaar duidelijk instructies van Ronny, maar als die dag of van gister toch nog iets uitliep, en dat ik ook nog loop te klagen dat een klant staat te wachten voor andere eisen/wensen. Ik denk dat we daar wel hulpmiddelen in kunnen gebruiken, om het te realiseren om het beter te doen.

Op dit moment wordt er vanuit de ontwikkeling puur reactief gewerkt. In mijn ideale wereld brengen de programmeurs zelf de klanten.

Samenvattend, Ik denk dat we een methode moeten verzinnen waardoor de ontwikkelaar documenten van de BPM-er sneller begrijpen en weet wat voor vragen ze moeten stellen om het werk te kunnen aannemen. En daar iets getrainder in moeten raken. En qua planning heeft dat een programmeur weet wat de komende tijd van hem wordt verwacht. Als de programmeur hier meer bewust van is, dat het meer rust in zijn hoofd geeft.

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

Interview Ronny

Wat is volgens jouw het probleem bij builders & performers?
We zijn een jong bedrijf, en daarin zie je dat we tegen dezelfde problemen aanlopen als elk jong technologisch bedrijf. Het probleem is een klein groeiende organisatie. En dat kent verschillende facetten Aan de ene kant in de operatie zie je dat de mensen iets zijn aan het bouwen, en dat geven ze aan de klant, en de klant gaat er naar kijken en zegt dat het niet goed is. **En dat moet reworked worden. En rework kost het meeste**. En in financiering, beloning**. Worden we voor rework niet betaald**. Dus we zeggen dat we iets in bepaalde tijd opleveren en we leveren het op. Feitelijk zou het dan goed moeten zijn.

Wat is de oorzaak dat er telkens rework plaats vindt?
De oorzaak is, nou kijk de problemen in de wereld zijn niet eenduidig. Het probleem zit het in meerdere dingen. B&P neemt een klus aan, en wij zorgen voor een fixed price. In de commerciële omgeving waarin we opereren, zit nu eenmaal zo als je tegen een klant zegt: het kan een dag duren het kan vier weken duren, daar kunnen ze niet mee werken. Vroeger werkten dat wel. Of je zei: we gaan eerst het probleem onderzoeken, dan gaan wij een rapportje schrijven en dan pas zijn we in staat om een plannetje te maken en dan maken we iets en weten eigenlijk niet wat het is en dat gaat je in totaal 20 dagen kosten. En dan moeten het nog testen. Nou tegenwoordig is er geen klant die daarmee akkoord gaat. Nu hoopt de organisatie dat je in een bepaalde branch opereert en daardoor wel gelijk het antwoord kan geven. En zoals je hebt gezien onze business is niet branch gebonden. En daardoor kunnen we dus heel veel diepe oplossingen maken**. Maar is het zo als de klant iets vraagt niet altijd precies weet waar hij gaat eindigen. Hij heeft wel een vraag en moet wat gebeuren bij hem, hij heeft daar ideeën over. En als we die ideeën zo 100% zouden uitvoeren, dan heeft niet altijd zijn doel bereikt**. En dan niet altijd bij het bouwen. Ze denken dan vaak laten we scrum doen, dan kunnen we het aanpassen. Maar het is niet alleen de bouw, mensen beginnen met denken hoe de software tot stand moet komen.

Maar dat is meestal niet het grootste probleem. Meestal is de eerste zet, de eerste oplevering van een project globaal genomen wel goed. En dan komt de praktijk erbij kijken en dan krijg je detaillering**. De klant ziet dan een probleem en die gaat zeggen: Ik wil dat hebben en soms wel en soms niet is de kwaliteit van zijn eigen observatie goed.** Soms vertelt die ons de oplossing van ik wil dat hebben en dan ben ik klaar. En soms kan hij ons alleen vertellen wat er mis gaat. En de tweede situatie is beter, want dan kunnen wij samen de oplossing bedenken. Wat vaker het geval is dat de klant denkt, jah als ik hun laat denken kost het mij geld. Dus als de klant denkt dat B&P alleen gaat bouwen dan ben ik wel goedkoper uit. En ze denken dan, het is ook in een keer klaar. Ik doe het denkwerk wel en laat hun maar bouwen. En de kwaliteit daarvan de varieert. Los van dat de kwaliteit varieert, is het mensen werk en een van de belangrijkste dingen die je hier hebt gezien, is dat jij hier binnen 2 weken al verschillende discussies hebt meegemaakt.

De discussie gister aan tafel, het ging alle kanten op, het ging nergens over. Maar je ziet op dat moment, human behaviour. Dan zegt de programmeur van dit zijn mijn oplossingen. En dan zegt de ander van jah nee wat wil je nou eigenlijk bereiken. En dat is iets wat heel vaak gebeurd. En dat heeft niet met programmeurs te maken maar met mensen. Nou als je dat zou kunnen verslechten, dan zou jij in een keer kunnen goed opleveren. Dus dan kijken we weer helemaal terug naar de watervall

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

methode, dan zou je dat bereiken. Maar de watermethode kan je niet meer in deze tijd toepassen. Dat is van 1980. En in onze economische en dynamische wereld waarin we nu zijn: alles moet snel snel snel. En dan denken ze, watervalmethode, rot jij maar een eindje op. En ergens daar tussen, moeten we optimum vinden.

Een balans tussen agile en waterval?
Jah, en dan dacht ik van we hebben een prototype tool die hiervoor goed gebruikt kan worden. Ooit observeerde ik jouw zelfde probleem, en toen zag ik dat de gebruiker heel academisch ging vertellen wat er aan de hand was en wat die wilde. En de efficiëntie waarmee dat gebeurde was echt verschrikkelijk. En dan zie ik daar er heel veel wordt gesproken en dan kijk ik naar de ontwikkelaars van dat ze het niet helemaal begrepen. Dus ik dacht, ik maak een prototyping tool waarmee je heel snel schermen redelijke wijze in elkaar kan zetten. Bedrijfsprocessen kan neerzetten. Dan heb je samen met elkaar beeld. En dat kan je met elkaar in discussie gaan en in de details gaan.
Werd volgens jouw het probleem hierdoor verminderd?

Ja, dat was het idee van die tool. **Het idee dus dat je geen ingewikkelde methodieken neerzet. Geen ingewikkelde schemas**. Maar gewoon wat het echt gaat worden. Dus in redelijke wijze in elkaar kan schetsen. Dus niet heel erg in details gaat zitten, maar dat was het idee achter het prototype tool. Maar ergens land dat niet. Het land niet in de organisatie, in de aanpak en je ziet bijvoorbeeld van een organisatie dat de RFC's zo klein zijn dat ze de tijd niet meer nemen om dat te doen. Of de volume van de RFC is zo groot dat je ze niet kan doorspreken. Ergens moet dus een balans moeten worden gevonden hoe je met dit middel omgaat. En misschien is het ook nog zo dat er meer bimmers op de vloer moeten komen.

Hier moeten we nu overdenken, we hebben gereedschappen, infrastructuur, ervaring en klanten. En hoe gaat we dit nou tot een geoliede machine smeden. En dat is waar jouw project zijn entree doet. En ik heb het antwoord nog niet, anders had ik jouw ook niet nodig.

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

Interviewboekje

Eerste interviewvragen

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

- Hoe verloopt het proces vanaf het moment dat een klant een eis/wens heeft tot aan de oplevering?
- Hoe vind je het proces momenteel verlopen?
    - Communicatie
    - Proces
- Vindt je dat het proces momenteel efficiënt verloopt?(gebruik hier de tabellen)
    - Waarom wel, waarom niet
- Welke problemen kom je in het proces tegen?
    - Wordt de eis/wens van de klant vaak begrepen?
        - Waarom wel, waarom niet, wat gaat er fout?

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

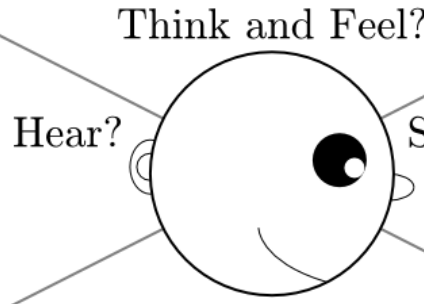
- Wat vind je momenteel van de geleverde eisen/wensen van klanten(of de specificatiedocumenten)
    - Wat mist er in het document
        - Denk aan requirements, processen, schermafbeeldingen
- Weet je van te voren wat je te verwachten staat?
    - Je verantwoordelijkheden?
- Vind je dat het proces momenteel goed wordt gemanaged, wat kan er anders/beter?
- Welke methodiek wordt er in de organisatie gehanteerd, en hoe vind je dat deze methode wordt gebruikt?
- Wat vind je van de tooling die nu wordt gebruikt?(denk aan eventum)
- Hoe zou jij het graag anders willen zien?

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

Business Model - The Empathy Map

Author:

Date: Iteration:



PAIN

GAIN

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

Wat gaat goed? Wat behouden?	Toelichting	Rol van medewerker zelf

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

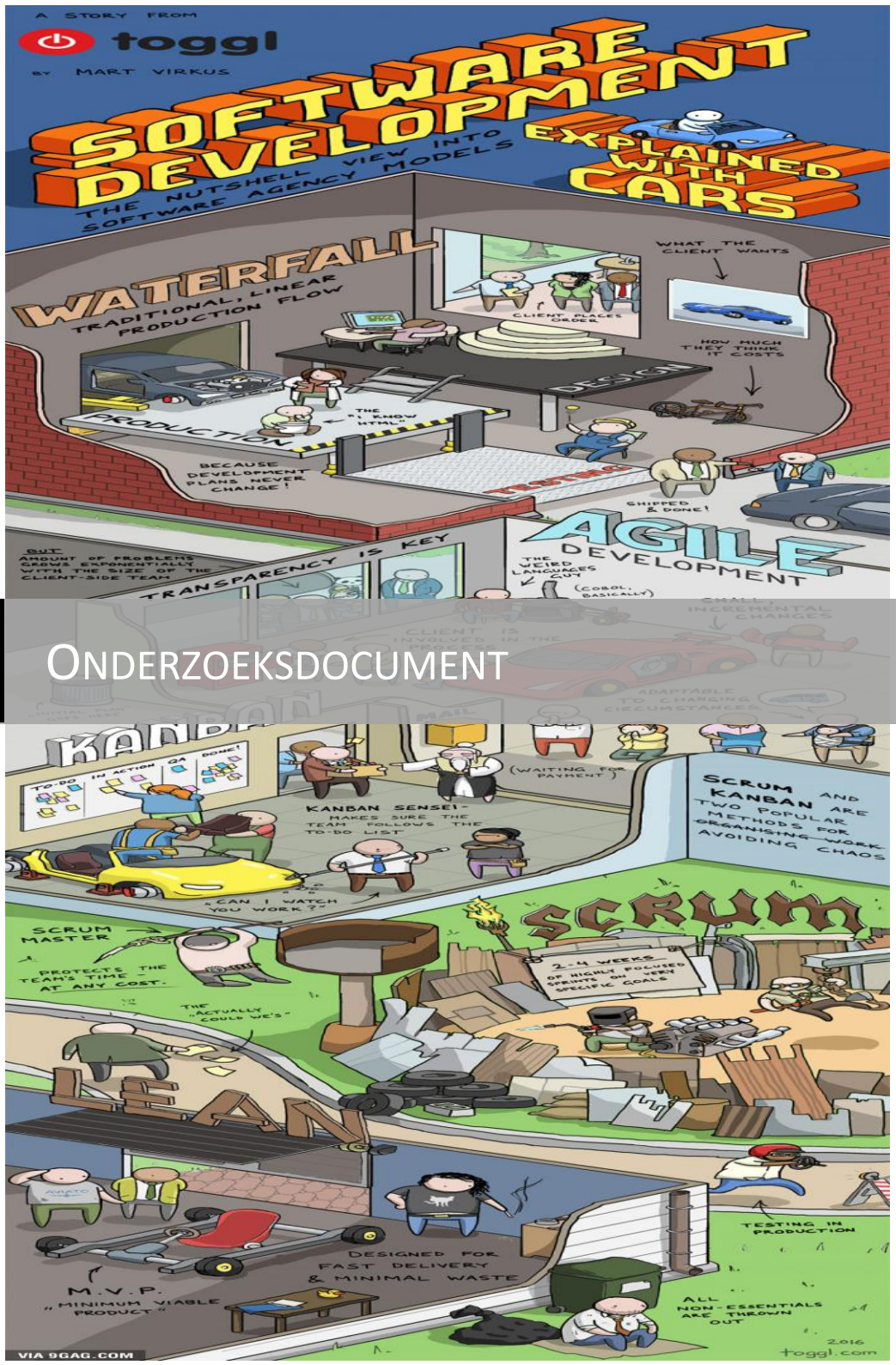
Knelpunt	Toelichting, oorzaken, gevolgen	Rol van medewerker zelf

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

Hoe zie je het proces nu? (teken het)

Builders en Performers
Sleepboot 9A, 3991 CN Houten
KVK: 5693946
IBAN: NL86INGB0006728623

16-3-2016



BUILDERS & PERFORMERS

ONDERZOEKSDOCUMENT

Onderzoek naar de verschillende software ontwikkelingsmethodes | Ruben van Stralen

Inhoud

1 Inleiding

Voor Builders & Performers is een literatuuronderzoek uitgevoerd over de verschillende technieken en methodes die gebruikt worden bij het softwareontwikkelingsproces. Daaruit is gekeken welke methode en technieken aansluiten bij de gewenste situatie voor B&P.

Het onderzoek heeft niet als doel om te analyseren wat de beste methodes/technieken zijn op de markt, wel naar welke technieken/methodes het beste aansluiten op de huidige situatie en de wensen van organisatie Builders & Performers.

In het document wordt per software ontwikkelingsmethode beschreven wat het is, hoe het proces verloopt, uit welke principes het bestaat, welke tools ervoor worden gebruikt en wat de voordelen en nadelen binnen het proces zijn. Verder is er ook een enquête gestuurd naar en geanalyseerd van verschillende organisaties over het gebruik van software-ontwikkelingsmethodes. Tot slot is er beschreven welke technieken er gebruikt kunnen worden om de eisen en wensen van klanten efficiënter over te brengen naar de ontwikkelaars.

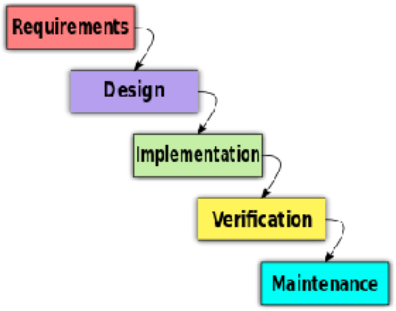
2 Definitie

Om te zorgen dat alle betrokkenen van het project hetzelfde beeld hebben van de begrippen, zijn hieronder de fundamentele termen van het onderzoek vastgesteld.

2.1 Software ontwikkelingsmethode

"Een software ontwikkelmethode is een methode die gebruikt wordt bij het ontwikkelen van software. Dit wordt ook wel softwarelevenscyclus of softwareproces genoemd." (Wikipedia, 2015)

2.2 Waterval

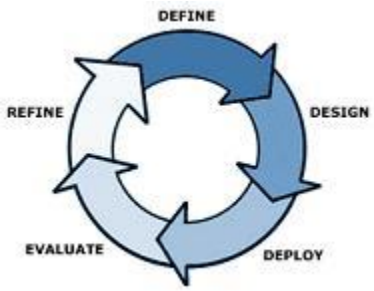


" Bij een watervalmethode worden alle requirements in een keer aan het begin van het traject opgesteld, goedgekeurd en opgenomen in de baseline." (de Swart, 2010)

"Waterfall methods take a set of (allegedly) clear and stable requirements, project how much the project will cost and, by using a sequential work breakdown structure (WBS), decide how long the project will take and how many resources it will require." (Nathan & Holte, 2016)

2.3 Iteratief

" Bij één iteratieve methode worden aan het begin van het traject alleen de globale requirements opgesteld, goedgekeurd en opgenomen in de baseline. Het detailleren van de requirements gebeurt voorafgaand aan de iteratie. Deze requirements worden tijdens de iteratie geïmplementeerd. " (de Swart, 2010)

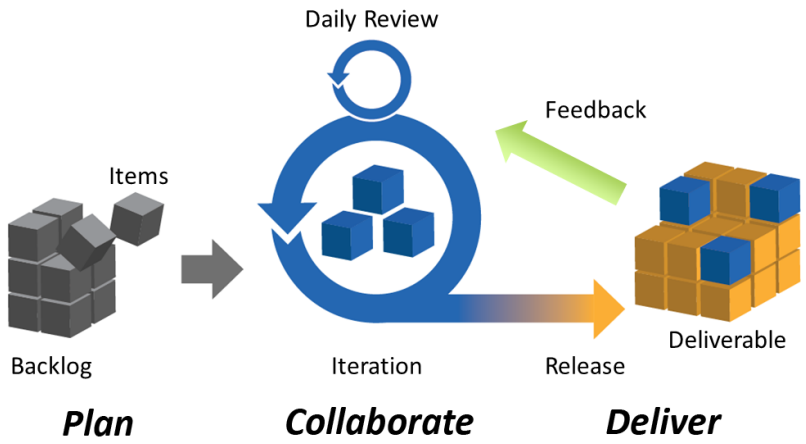


"Iterative methods build on incremental waterfall by adding an explicit opportunity for feedback after each increment or iteration. This allows for the early detection and resolution of problems in the requirements. Done correctly, the iterative approach assumes that requirements cannot be frozen before construction begins. Known requirements are realized and implemented using short cycles of analysis design and implementation, enabling the system to evolve. (Nathan & Holte, 2016)

2.4 Agile

"Bij een Agile methode worden voorafgaand aan de iteratie alleen de globale requirements goedgekeurd en opgenomen in de baseline. Detaillering en implementatie van de requirements vindt tijdens de iteraties plaats." (de Swart, 2010)

"Agile methods are fixed-time, fixed-resource methods that plan delivery based on function decomposition rather than time sequence). Agile projects do not require a clear set of requirements at the beginning, as the feedback from the customer is used to determine the best solution during the project. Because of the need for constant feedback, the timeboxes here are shorter than for iterative (usually two to four weeks). Such rapid feedback loops allow this functionality to easily handle Mode 2 projects, where the solution is not known upfront. This class of methods may also include lean/Kanban approaches, which don't focus on iterations but do share the same planning and cultural needs. " (Nathan & Holte, 2016)



3 Is de organisatie klaar voor agile en welke methode is dan geschikt?

Agile is een methode die in veel IT-organisaties is ingevoerd. Steeds meer organisaties wijzigen van een watervalachtige methode naar Agile methode. Echter, niet alle organisaties zijn klaar om met de gedachtegang van agile te werken. Een organisatie kan veranderen in een agile omgeving als er een cultuur ontstaat van een samenwerkend teamverband. Dit houdt in dan dat:

- Team georiënteerd werken, en niet individueel
- Veranderingen geaccepteerd worden
- Doel georiënteerd werken, dus niet taak georiënteerd
- Vertrouwen

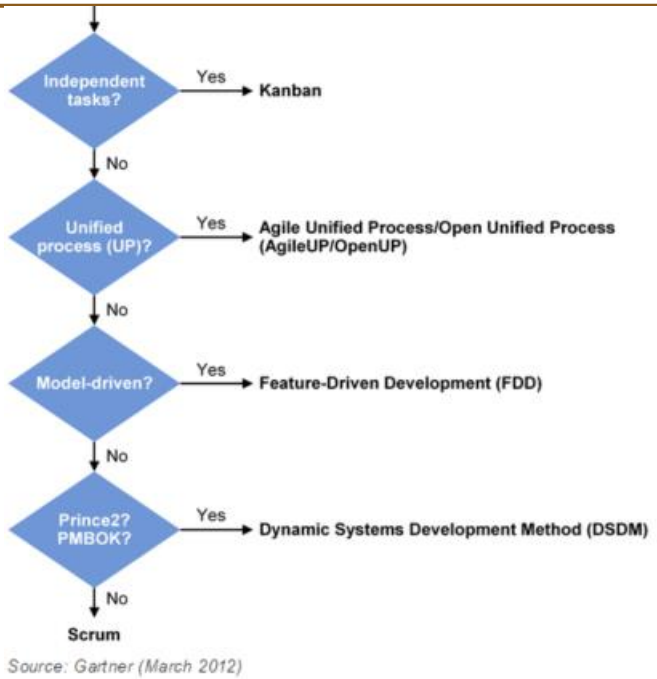
(Wilson, 2012)

De eerste stap die gezet moet worden om over te stappen naar een Agile principe, is culturele verandering. De verandering moet zorgen dat we niet denken als individu, maar als team. Om naar agile te veranderen, moet niet de focus gelegd worden op processen en technologie, maar op de medewerkers en de cultuur. De meeste organisaties starten met een pilot fase. Om een pilot fase te starten moet er wel gedacht worden aan de volgende punten (Hotle & Norton, 2010):

- Het ontwikkelde product moet niet belangrijk zijn voor de business.
- Het product moet binnen drie en zes iteraties “live” zijn.
- Een proces coach moet beschikbaar zijn om het proces te ondersteunen en feedback te geven.

Nu is het nog de vraag welke agile methode geschikt is, want er zijn namelijk meer dan 15 verschillende methodes op dit gebied. Om een methode te kiezen, moet je niet proberen om de perfecte te kiezen volgens de markt, maar een methode die past bij de organisatie. Maak hiervoor een decision tree om te selecteren welke methode goed genoeg is voor de organisatie. Daarnaast is het belangrijk dat je voldoende training en kennis hebt op het gebied van de gekozen methode. Ook dien je rekening houden dat iedere ontwikkelaar enige ervaring heeft met de methode.

Gartner heeft een eigen decision tree ontworpen hoe een organisatie kan kiezen voor een agile methode.

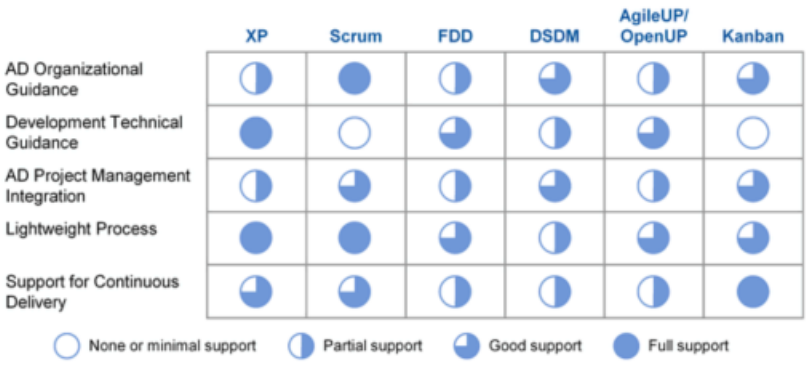


Source: Gartner (March 2012)

Voor organisaties die met veel onafhankelijke taken werken, is kanban een prima agile methode om mee te starten. Bij kanban worden de taken in het product backlog geplaatst waarbij de taak met de hoogste prioriteit als eerste wordt verwerkt. Het zorgt ervoor dat er in een lopende band wordt gewerkt.

Voor organisaties die werken met een een sterk Unified Process, kunnen ze overstappen naar een lichtere variant, namelijk AgileUP. Hierbij geldt dezelfde stappen als bij RUP namelijk, inception, elaboration, construction en transtion. Alleen bestaat AgileUP uit minder artefacten, rollen en activiteiten.

Voor organisaties die erg design gericht werken, is FDD een uitstekende methode om mee te beginnen. Voor organisaties die willen blijven werken met Prince2 of PMBOK maar toch iets meer richting agile willen gaan, dan past DSDM (Wilson, 2012).



De organisatie die overblijven kunnen overstappen naar Scrum. Scrum is een populaire methode en er is wereldwijd genoeg informatie over te vinden hoe je de methode in de organisatie kan toepassen.

Extreme Programming(XP) kan ook gebruikt worden in plaats van Scrum. XP is een goed model als het gaat om efficiënt te coderen, echter biedt het weinig ondersteuning aan projectmanagement.

(Wilson, 2012)

De uiteindelijk gekozen methode moet niet in een klap geïmplementeerd worden. Middels kleine stappen kan een methode succesvol geintegreed worden in de organisatie. Zorg dat je een techniek per keer toepast op het project (Wilson, 2012).

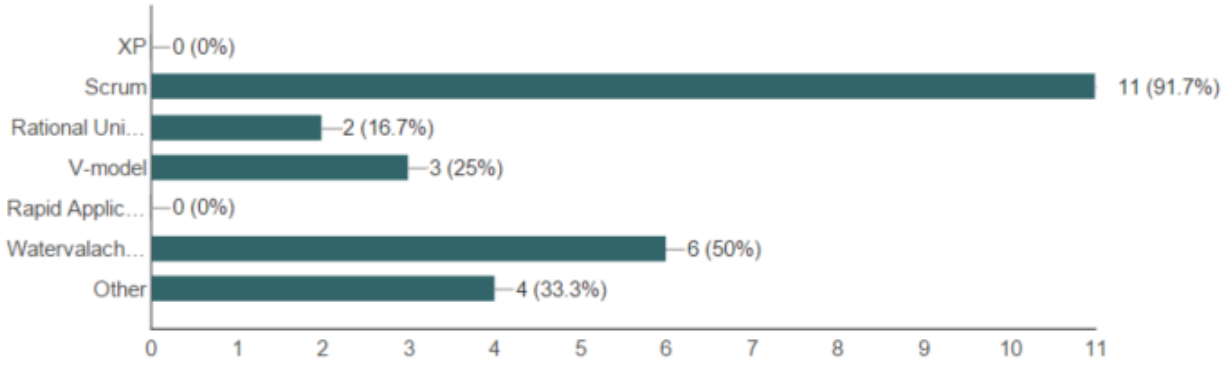
4 Analyse van de enquêtes over software ontwikkelingsmethode

Om een indruk te krijgen hoe momenteel de markt omgaat met de verschillende software ontwikkelingsmethodes, zijn er enquêtes verstuurd naar verschillende organisaties. De respons op de enquêtes was te laag om een representatief beeld te creëren. De analyse op de enquêtes wordt daarom alleen gebruikt om een eerste indruk te krijgen hoe andere bedrijven werken met een software ontwikkelingsproces.(in bijlage A staan alle resultaten)

De respons kwam uit allerlei verschillende vakgebieden in de ICT, enkele voorbeelden; (technische) consultants, ontwikkelaars, testers en managers. Ook kwamen zij uit allerlei verschillende organisaties zoals, bol.com, belastingdienst, mk2 software en HSO. De afdelingen hadden verschillende groottes, sommige organisaties werkten in teams van 5/10 medewerkers, andere rond de 50 medewerkers en twee andere organisaties hadden een grootte van meer dan honderd man. Gemiddeld kwam de afdelingsgrootte op 39 medewerkers.

De eerste vraag die in de enquête werd gesteld was of de organisatie gebruik maakten van een softwareontwikkelingsproces. 71,4 % maakten wel gebruik van een software ontwikkelingsmethode in de organisatie. Organisaties die geen gebruik maakten van een software ontwikkelingsmethode hadden voor 83% geen behoefte aan een software ontwikkelingsmethode.

De organisaties die wel gebruik maakten van een software ontwikkelingsmethode hanteerde de volgende methodes:



(XP, SCRUM, RUP, V-model, RAD, waterachtige methode, overig)

Scrum wordt bijna in alle organisaties toegepast. Op deze vraag waren ook meerdere keuzes te selecteren. De organisaties waar de afdeling groter was dan 35 medewerkers, maakten gebruik van zowel een watervalachtige/iteratieve methode(als RUP, V-model) in combinatie met scrum. Uit een enquête uit andere bron kwam ook al naar voren dat 53% van de organisaties een waterval methode combineerde met een agile methode (Starke, 2013).

Ook werden er twee mindere populaire methodes opgenoemd die de organisaties gebruikten, genaamd Safe Agile en Side-by-Side development.

De principes van de methodes werd door iedereen begrepen. De regels van de methode werden voor 75% nageleefd.

Vervolgens is gevraagd of de gehanteerde methode efficiënt werkt binnen de afdeling. Daaruit zijn twee richtingen gekomen. De organisaties die een methode volledig hebben geïntegreerd in de organisatie, zeggen dat de methode efficiënt werkt, niet alleen voor IT, maar ook voor de business zelf. Alleen bij organisaties waar de methode niet volgens de regels wordt gehanteerd of niet volledig is doorgevoerd, wordt de methode als inefficiënt genoemd.

Hoewel 45% heeft aangegeven dat een methode inefficiënt werkt in de organisatie, geeft 83% procent van de ondervraagde aan geen voorkeur te hebben aan een andere methode. Waarschijnlijk zijn organisaties pas begonnen met het invoeren van een agile(scrum) methodiek, het moet nog een plek in de cultuur van de organisaties krijgen en dat heeft nog tijd nodig. Hier is geen verder onderzoek naar gedaan.

Verder is er gevraagd welke tools de organisaties gebruikten om de software ontwikkelingsmethode te ondersteunen. Twee organisaties maakten geen gebruik van tooling. Hieronder is een lijst opgesteld met de tools die worden gebruikt:

- JIRA(3 keer gebruikt)
- Fysieke scrumbords
- Visual Studio Online(2 keer gebruikt)
- TOPdesk(2 keer gebruikt)
- Icescrum
- Team Foundation Server(2 keer gebruikt)
- Excel
- Connect People
- Target Process

5 Analyse enquêtes van Ambysoft

In het vorige hoofdstuk is een analyse gedaan van de opgestuurde enquêtes over software ontwikkelingsmethodes. De respons daarop was te weinig om een representatief beeld te schetsen. Om die reden is er gezocht naar bronnen die een soort gelijke enquête hadden gehouden. De website Amysoft opgericht door de beroemde Canadese softwareontwikkelaar Scott Ambler houdt jaarlijks enquêtes op dit gebied. Hieronder is een analyse gedaan van de informatie uit de enquêtes.

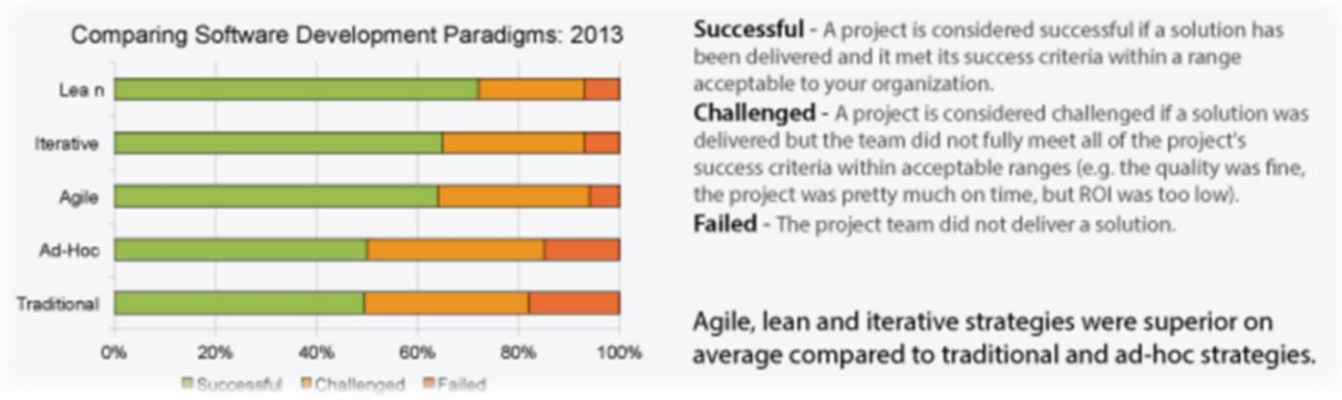
5.1 Succes van software ontwikkeling paradigma's

Als het gaat om software ontwikkeling paradigma's zijn er vijf grote namen, namelijk:

- Lean
- Iteratief
- Agile
- Ad-hoc
- Traditioneel(waterval)

In de enquête worden lean, agile en iteratieve paradigma's beschouwd als de meest effectieve (zie afbeelding). Qua productkwaliteit heeft lean een ruime voorsprong op de andere paradigma's. Als het gaat om de waarde die het oplevert voor stakeholders en de effectiviteit van het plannen staan zowel lean als iteratief vooraan. Op ROI worden Agile en lean erg effectief gezien.

http://www.ambysoft.com/surveys/success2013.html



5.2 Requirements

In de eerste fase van elk project is het fundamenteel om de requirements van de eindgebruikers vast te leggen en te valideren. Verschillende methodes kunnen worden gebruikt om de requirements te achterhalen van de organisatie. De methodes die vaak gebruikt worden om requirements te achterhalen zijn:

- Houden van interviews
- Demonstratie van werkende software
- Op een whiteboard schetsen

Technieken die weinig worden gebruikt zijn gestructureerde modellen, CASE tool modelleren en JIT modelleren.

Om de requirements vast te leggen wordt er vaak gebruik gemaakt van:

- Use cases
- Free-form diagrammen
- Business rules

Technieken die zelden worden gebruikt zijn class diagrammen en mind maps.

http://www.ambysoft.com/surveys/stateOfITUnion201105.html

5.3 Agile practices

Elke methode maakt gebruik van verschillende technieken, regels en waardes voor een efficiënt verloop in het softwareontwikkeling proces. In deze enquête is geanalyseerd naar de effectiviteit van de best practices. De resultaten hieronder komen van 123 responsen en komen grotendeels uit een team van 1-10 medewerkers.

Pair programming(uitgelegd in hoofdstuk 6.1), Active Stakeholder Participation, Potentially Shippable Software en Burndown Tracking worden niet erg effectief genoemd in het proces.

De Agile practices die wel effectief in de organisatie zijn continu integratie(uitgelegd in hoofdstuk 6.1), daily stand up meetings(uitgelegd in hoofdstuk 6.2) , developer TDD(uitgelegd in hoofdstuk 6.6), iteraties plannen(uitgelegd in hoofdstuk 6.2).

Iteraties plannen, daily stand up meeting en continu integratie worden ook als eenvoudige technieken benoemd om snel te leren. Hoewel TDD als effectieve methode wordt benoemd, is het volgens de enquête wel de lastigste methode om te leren.

http://www.ambysoft.com/surveys/practices2009.html

6 Hybrid waterval-agile

Alle organisaties worstelen met de vraag of ze de filosofie moeten volgen van agile of blijven bij de oude paradigma waterval. Agile wordt gezien als een flexibel model als het gaat om plannen, feedback van de klant verwerken, bouwen en testen. Agile streeft er naar voor meer interactie met de klant en de communicatie met de ontwikkelaars onderling. Ook reageert Agile op de veranderingen van de eisen en wensen van de klant.

Waterval heeft het voordeel dat de requirements, scope en budget is vastgezet. Bij zo'n project weet de organisatie wanneer het project start en wanneer het project eindigt. Ook gaat waterval ervan uit dat bijna alles wordt gedocumenteerd en vastgelegd. Hierdoor verdwijnen de onduidelijkheden in een project.

Beide paradigma's hebben ook valkuilen. Agile projecten zijn namelijk lastig te voorspellen, zowel qua planning als budget. Ook vraagt agile een hoop interactie met de klant, niet elke klant heeft hiervoor tijd. Als de klant weinig interactie heeft met de ontwikkelaars, dan mislukt het agile project. Waterval is niet flexibel. Als de requirements eenmaal vast staan, is er geen fase terug om de requirements te vernieuwen. Als er iets anders wordt opgeleverd dan wordt gevraagd, dan is er veel rework nodig om gewenste systeem op te leveren.

Echter is het ook mogelijk om beide paradigma's te combineren. Zoals in de enquête werd vastgesteld, gebruiken veel organisaties een combinatie van agile en waterval methodes. Binnen de IT wordt er dan ook wel gesproken van hybrid-waterval-agile methode. De hybride paradigma haalt het beste van beide methodes naar voren. Hoewel er in de literatuur verschillende benaderingen en technieken worden gebruikt om de hybride paradigma te beheersen, wordt meestal het volgende proces gebruikt (Guay, 2015):

In de eerste fase worden alle business en gebruikers requirements geanalyseerd. Daarna wordt een High level design gedocumenteerd voor het project. Daarin staan alle requirements, architectuur, scope, flowdiagrammen, wireframes, ect. In andere hybride agile-waterval methode wordt dit document niet in de opgenomen, maar als de iteratie start, wordt eerst een design gemaakt en daarna gebouwd. Alle requirements worden vervolgens in het product backlog vastgelegd. Nu begint de fase van het agile proces. Voordat een iteratie start, wordt eerst gekeken welke requirements de hoogste prioriteit heeft. Daarna worden de requirements gebouwd, getest en met de klant gereviewd. Als de eisen en wensen ontbreken of niet volledig zijn gebouwd, dan wordt dit in de volgende iteratie gecorrigeerd. Deze cyclus wordt in een aantal iteraties gedaan, totdat de opgestelde requirements zijn geïmplementeerd. In de laatste fase wordt er met de klant een acceptatietest uitgevoerd van het gehele systeem.

De hybride waterval-agile methode laat het beste van beide methodes zien. De methode zorgt ervoor dat het project vooruit gaat binnen de opgestelde tijd en budget, en tegelijkertijd rekening houdt met de interactie met de klant via de korte iteraties.

(Heuer, 2015)

7 Een methode in de organisatie adopteren

Veel organisaties willen een methodiek introduceren om het huidige software ontwikkelingsproces te verbeteren. Echter wordt er veel weerstand geboden door de ontwikkelaars. Dit heeft als oorzaak dat de ontwikkelaars de methodiek niet als nuttig beschouwen. Hoewel een methode niet als een wondermiddel moet worden beschouwd om problemen op te lossen, heeft CMM (Capability Maturity Model) onderzocht dat een methodiek in volwassenstatus erg effectief blijkt te zijn. Een volwassen geïmplementeerde methodiek zorgt ervoor:

- Dat de druk niet wordt gelegd op de ontwikkelaars, maar op het proces zelf.
    - Daardoor worden de vaardigheden van de ontwikkelaar meer uitwisselbaar, en dat is erg belangrijk door het dagelijkse te kort en hoge omzet van de ontwikkelaars.
- Verbetert de productiviteit.
- Door de documentatie te verstrekken en bij te houden, wordt rework gereduceerd.
- Een hogere volwassenheid zorgt voor een betere kwaliteit van de software en functionele requirements hebben een betere match met de oplevering.
- Vermindert tijd, inspanning en kosten.

Een volwassen methodiek heeft zoals bovengenoemd vele voordelen. Toch wordt er veel weerstand geboden voor het introduceren van een methodiek bij de ontwikkelaars.

Door onderzoek is geconstateerd dat een methode door ontwikkelaars wordt geaccepteerd door:

- Nuttigheid
- Simplistisch
- Compatibiliteit
- Vrijwilligheid
- Subjectieve norm

Nuttigheid is de belangrijkste reden dat een methode door de ontwikkelaar wordt geaccepteerd om te gebruiken. Hierbij wordt verstaan dat de methode bij de persoon zelf zorgt voor meer productiviteit, kwaliteit van het eigen werk verbeterd wordt en het werk gemakkelijker maakt om uit te voeren.

De methode moet ook eenvoudig zijn. De methode moet gemakkelijk te begrijpen zijn door de gehele organisatie en ook eenvoudig te gebruiken zijn. De methode mag niet veel tijd nemen van de dagelijkse werkzaamheden.

Andere reden om een methode te adopteren is compatibiliteit. De methodiek moet voldoen aan de aspecten op de manier hoe een ontwikkelaar te werk gaat. De methode moet aansluiten op de manier hoe ze werken.

Daarnaast speelt vrijwilligheid een rol bij het invoeren van een methodiek. Hiermee wordt bedoeld dat een ontwikkelaar de methode vrijwillig aanpakt en niet wordt gedwongen om het te gebruiken, dit kan namelijk averechts werken.

Als laatst wordt de subjectieve norm als onbewuste reden genoemd dat een ontwikkelaar de methode adopteert. Sociale invloed speelt een enorme rol hierin. Als de ontwikkelaar een methode als nuttig en bruikbaar beschouwt, maar andere medewerkers en managers denken daar anders over, dan is de kans groot dat de ontwikkelaar een methode niet gaat gebruiken. Daarom is het belangrijk dat de omgeving de methode accepteert, waardoor ook de andere ontwikkelaars de methode gaan accepteren.

Een methode invoeren heeft volgens het CMM wel degelijk een belangrijke waarde om de effectiviteit van het software ontwikkelingsproces te verhogen. Hoe volwassener de methode wordt gebruikt in de organisatie, hoe meer effectiviteit de methode oplevert. Echter biedt het invoeren van een methode weerstand bij de ontwikkelaars, dit kan worden vermijdt om rekening te houden met de nuttigheid, eenvoudigheid, compatibiliteit, vrijwilligheid en de onbewuste subjectieve norm.

(Riemenscheider, Hardgrave, & Davis, 2002)

8 De software ontwikkelingsmethodes

Hieronder is een lijst opgesteld van de gebruikte software ontwikkelingsmethodes op de markt:

8.1 eXtreme Programming

Achtergrond

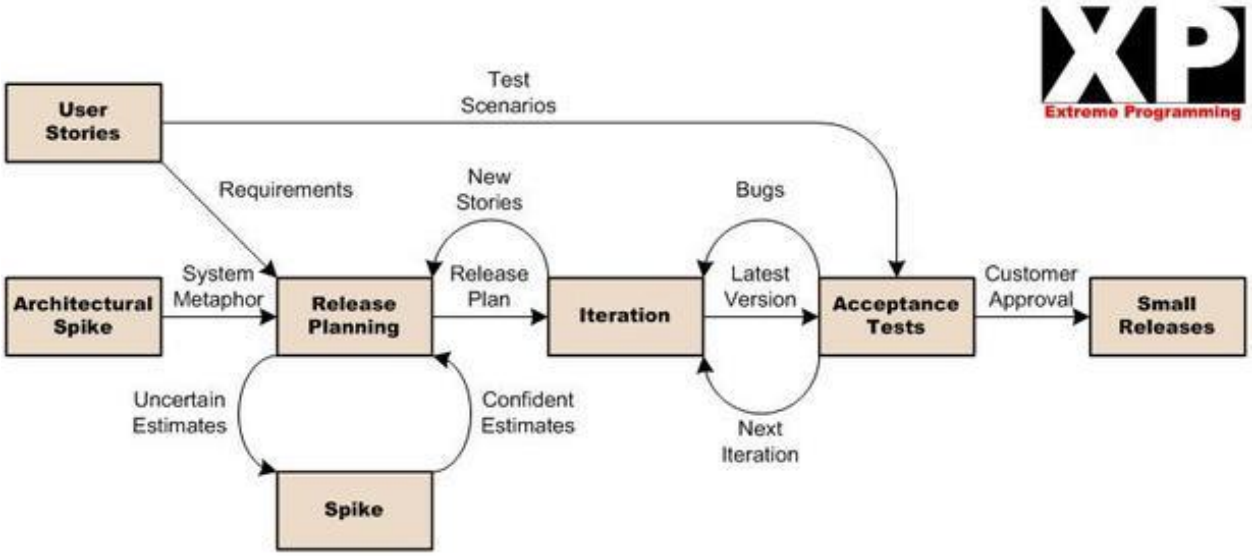
eXtreme Programming(XP) is een vorm van een agile ontwikkelingsmethode, dat vanaf de bottom-up wordt uitgevoerd. De methode focus zicht op vier onderdelen, namelijk: feedback, continu proces, gedeelde kennis en het welzijn van de ontwikkelaars. Het voornaamste doel waar XP zich op focust is dat alles draait om de code bij het maken van software. Hiermee wordt bedoeld dat de code door ontwikkelaars wordt gereviewd, dit wordt ook wel pair programming genoemd.

Vaak wordt gedacht dat daarom deze methode teveel onnodige tijd verspilt. Echter, onderzoek heeft aangetoond dat peer-to-peer review een krachtiger middel is om je te bewapen tegen bugs. Zelfs meer dan systematische testen.

Om de methode succesvol te integreren in de organisatie, moet je zorgen voor dat de organisatie niet bang is voor een stijgend aantal arbeidsuren en rekening houdt met grote weerstand van ontwikkelaars. (Norton, Agile Foundation: Extreme Programming, 2010)

Proces

XP bestaat uit vier kern activiteiten; coderen, testen, luisteren en ontwerpen. Dat wordt uitgevoerd door vijf rollen; ontwikkelaar, klant, tester, tracker en coach. Iteraties in dit proces is essentieel.



Net zoals bij de meeste methodieken worden eerst de requirements van de gebruiker opgesteld. XP doet dit met door gebruik van user stories. De user stories worden in dit traject niet alleen gebruikt om door de ontwikkelaar te laten bouwen, maar ook als input voor de acceptatietest. Ook wordt er in het proces spike oplossingen beschreven om technische en design problemen te voorkomen. Daarna wordt er een release planning gemaakt om iteraties te creëren. Er wordt bij elke user story bepaald hoelang ze daarmee bezig zijn. Als user stories zijn gebouwd in de iteratiefase worden ze in de acceptatiefase getest door de klant. Als er bugs worden gevonden, worden deze in de volgende iteratie weer verwerkt. Na de acceptatietest wordt een small release uitgevoerd bij de klant. Vervolgens wordt er gewerkt aan de volgende iteratie (Wells, 2013).

Principes

Voor XP geldt twaalf principes:

- Planning Game: Bepaal op korte termijn de scope voor de volgende release gecombineerd met business prioriteiten.
- Small Releases
- Metaphor: Bespreek in een simpel verhaal met het gehele ontwikkelingsteam hoe het systeem gaat werken.
- Simple Design: Systeem moet op simpele mogelijke worden ontwikkeld. Als er extra complexiteit in wordt ontdekt, maar dat worden verwijderd.
- Testing
- Refactoring: Ontwikkelaars herstructureren het systeem zonder dat het gedrag van het systeem wordt veranderd, om te zorgen voor verwijdering duplicatie, verbetering communicatie, eenvoudiger of flexibeler.
- Pair Programming: Alle code is beschreven door twee ontwikkelaars.
- Collective Ownership: Iedereen kan de code aanpassen.
- Continuous Itegration: Integreer en bouw het systeem meerdere keren op een dag, elke tijd een taak is geïmplementeerd.
- 40-Hour Week: Werk niet meer dan veertig uur per week.
- One-site Customer: Zorg voor een klant die altijd bereikbaar is om te antwoorden op vragen.
- Coding standard: Ontwikkelaars schrijven de code volgens een set of regels van best practices.

(Runeson & Greberg, 2008)

Waardes:

XP is niet alleen een set of regels, maar een manier om in harmonie te werken aan je persoonlijke en zakelijke waardes. XP beschrijft er vijf:

Eenvoudig: *"We will do what is needed and asked for, but no more. This will maximize the value created for the investment made to date. We will take small simple steps to our goal and mitigate failures as they happen. We will create something we are proud of and maintain it long term for reasonable costs. "*

Communicatie: *" Everyone is part of the team and we communicate face to face daily. We will work together on everything from requirements to code. We will create the best solution to our problem that we can together."*

Feedback: *" We will take every iteration commitment seriously by delivering working software. We demonstrate our software early and often then listen carefully and make any changes needed. We will talk about the project and adapt our process to it, not the other way around."*

Respect: *"Everyone gives and feels the respect they deserve as a valued team member. Everyone contributes value even if it's simply enthusiasm. Developers respect the expertise of the customers and vice versa. Management respects our right to accept responsibility and receive authority over our own work."*

Lef: *"We will tell the truth about progress and estimates. We don't document excuses for failure because we plan to succeed. We don't fear anything because no one ever works alone. We will adapt to changes when ever they happen. "*

(Wells, 2013)

Tooling

Voor XP is een tool FIT(Framework for Inegrated Test) geïntroduceerd. De software zorgt voor samenwerking en communicatie. FIT heft als doel dat alle betrokkenen leert hoe de software zou moeten werken en hoe het echt werkt. http://fit.c2.com/

Voordelen en nadelen

De voordelen:

- Zeer succesvolle projecten bekend
- Interactief te werk gaan
- Vertrouwen in ontwikkelaar
- Continu proces verbetering
- Geen dure tools
- Flexibel
- Minder bugs
- Snel resultaten

De nadelen:

- Weinig ervaring op dit gebied
- Pair programming ligt alleen voor ontwikkelaar die ook werkelijk teamspeler zijn
- XP is niet altijd eenvoudig
- Heeft niet veel tools
- Wordt veel aandacht gevraagd van de klant
- Heeft geen specifieke artefacten gedefinieerd.

(Runeson & Greberg, 2008) (Al-Saleem & Ullah, 2015)

8.2 Scrum

Achtergrond

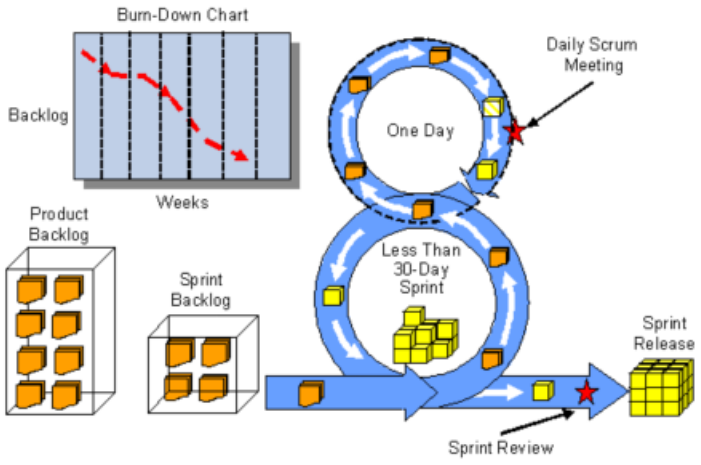
Scrum is ook een agile methode die wordt ingezet om flexibel te werk te gaan. Het is in principe geen aanpak, maar een filosofie die afscheid neemt van het blauwdruk denken. Scrum moet zorgen voor betrokkenheid, snelheid(snel werkende producten en communicatie) en duidelijkheid over de voortgang. Het scrumteam bevat meestal drie rollen:

Product-owner:	Dit is meestal de klant en opdrachtgever. Hij beheert het productbacklog bij. Met de product-owner is het dan ook belangrijk dagelijks te overleggen.
Ontwikkelteam:	Bestaat meestal uit ontwikkelaars. Zij zorgen voor ontwikkeling, testen en documentatie.
Scrummaster:	Hij begeleidt het team en zorgt dat het scrumproces wordt gevolgd.

Scrum maakt gebruik van drie artefacten, namelijk de product backlog, sprint backlog en de burndownchart (Norton, 2007).

Proces:

Het proces begint bij de stakeholders. Zij stellen een gehele lijst op met eisen en wensen aan het systeem. De product-owner prioriteert vervolgens de lijst welke requirements belangrijk zijn of niet. Deze lijst wordt vervolgens geplaatst in de product backlog. Vervolgens wordt er met het team overlegd welke eisen en wensen uit het productbacklog wordt meegenomen naar de volgende sprint. Een sprint kan bestaan uit 2-4 werken. Het sprintbacklog bestaat uit drie fases:



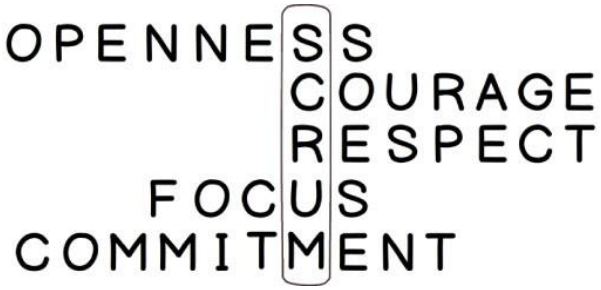
To do:	Wat er nog gedaan moet worden
In Progress:	Wat er al gedaan is
Done:	Wat is afgemaakt

Vaak wordt er dan ook gebruik gemaakt van een burndown chart. Hiermee kan je zien hoeveel user stories zijn gedaan en wordt voorspeld wanneer de sprint ongeveer opgeleverd wordt.

Elke dag wordt er met het team een daily standup gehouden. Hierin wordt aan ieder teamlid gevraagd:

- Wat heb je gister gedaan?
- Wat ga je vandaag doen?
- Tegen welke problemen loop je aan?

Als de sprint klaar is wordt er met het team een sprint review gedaan. Hierin wordt samen met de product-owner het product opgeleverd. Vervolgens wordt er met het team besproken wat er goed ging en wat beter had gekund (Norton, 2007).



Waardes

Net als XP bestaat scrum uit vijf waardes die cruciaal zijn om het proces vloeiend te laten verlopen (Verheye, 2013).

Toewijding:	Iedereen moet zich volledig inzetten.
Focus:	Alleen letten op wat er gaan moet worden in de sprint.
Openheid:	Van elkaar op de hoogte zijn wat er speelt.
Respect:	Spreekt voor zich.
Lef:	Durf vragen te stellen en met nieuwe oplossingen te komen.

Tooling

Op de markt zijn er honderden producten om het scrum proces te ondersteunen. De tools kunnen de user stories bijhouden, sprintbacklog maken, burndownchart bijhouden, ect. Hieronder is een lijst gemaakt met de bekendste tools die het gehele proces ondersteunen:

JIRA:	https://www.atlassian.com/software/jira
Trello:	https://trello.com/
Targetprocess:	https://www.targetprocess.com/
Tiaga:	https://taiga.io/

Voordelen en nadelen

Voordelen:

- Populaire methode, er is veel informatie over te vinden
- Veel trainingen en mentoren beschikbaar om het proces te begeleiden
- Genoeg tools om het proces soepel te laten verlopen
- Elk lid weet van elkaar waarmee ze bezig zijn
- Veel feedback momenten, waardoor het eenvoudiger is om nog te veranderen
- Is bewezen dat het zorgt voor een accurate schatting van de planning en inspanning
- Het gaat verder waar Extreme Programming is opgehouden
- Kan worden gebruikt in organisaties die binnen level drie zitten van het Capability Maturity Model

Nadelen:

- Hoewel het zorgt om projectmatig te werken, maar verschaft geen advies over het ontwikkelingslevel
- Voor organistisch die planmatig(watervalachtig) te werk gaan is het lastig om te switches naar de scrummethodiek
- Zorgt voor scope creep
- Als een teamlid niet is toegewijd, dan is er een grote kans van falen
- Framework is meestal alleen succesvol als de leden ook enige ervaring hiermee hebben

8.3 Kanban

Achtergrond

Kanban is een methode die gebruikt wordt bij concepten als lean manufacturing en just-in-time productie. Het is een manier om georganiseerde chaos te structuren, prioriteren en zich duidelijk richten op datgene waarmee ze bezig zijn.

Met kanban hoeft niet veel organisatorisch veranderd te worden. De huidige rollen, verantwoordelijkheden, processen en activiteiten kunnen behouden worden. Kanban beschrijft namelijk geen procedure hoe je het moet opzetten. Je kan langzame stappen zetten van hoe de organisatie nu werkt en hoe het in de toekomst wilt werken.

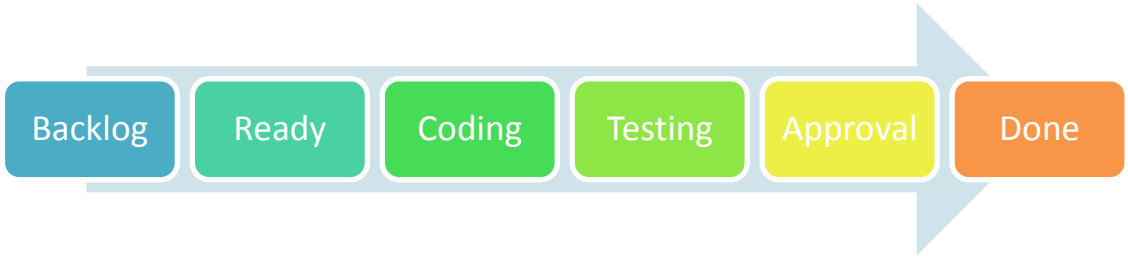
Principes

1. Start met wat je aan het doen bent
2. Je accepteert incrementele veranderingen
3. Respecteert de huidige situatie

4. Moedig leiderschap op alle niveaus aan

Proces

Zoals al staat beschreven is het proces geheel naar eigen keuze in te delen. Bij kanban worden eerst alle eisen/wensen van de klanten verzameld. De eisen en wensen worden vervolgens geprioriteerd. Degene met de hoogste prioriteit wordt als eerste verwerkt. De fases waarin de eis of wens wordt verwerkt, is naar eigen smaak in te richten. Meestal wordt het proces in deze volgorde ingedeeld:



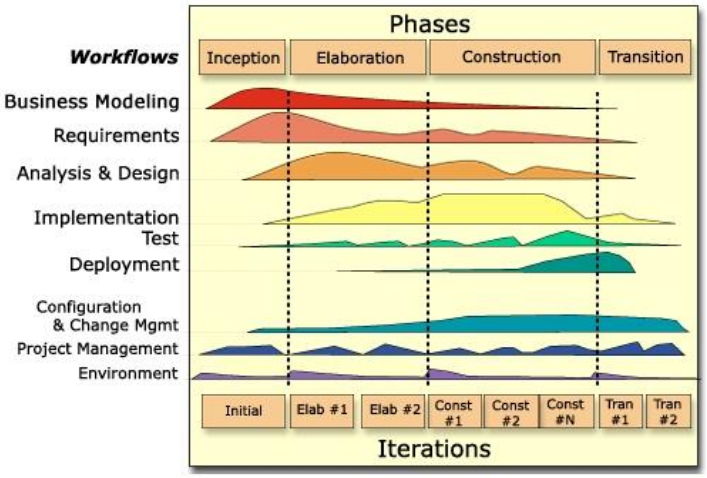
Tools

Voor kanban heb je alleen een bord en post-its nodig. Hiermee kan je het proces monitoren, beheren en wijzigen. Ook zijn er software tools beschikbaar als je geen gebruik wil maken van een fysieke kanbanbord. Dit zijn dezelfde tools die ook beschreven staan bij de scrummethodiek.

(Kanban, 2016)

8.4 Rational Unified Proces & Agile Unified Proces

Achtergrond



RUP (Rational Unified Proces) is een iteratieve manier om software te ontwikkelen. RUP is gecreëerd door de software gigant IBM. De methode bestaat uit vier fases, 9 disciplines, meerdere rollen en artefacten. RUP verhelpt de problemen die zorgde voor mislukte projecten, zoals AD-hoc requirements, onduidelijke requirements, overmatige complexiteit en onvoldoende testen (de Vries, 2013).

Proces

De methode wordt in vier fasen opgedeeld, iedere fase wordt vervolgens een of meerder iteraties. De fases bestaan uit:

Inception: Hierin bepaal je de scope van het project, de haalbaarheid en de business requirements.

Elaboration: Hierin worden de risico's vastgelegd, processen in kaart gebracht, prototypes gemaakt en de software-/gebruikers requirements opgesteld.

Construction: Hierin wordt in meerdere iteraties het gewenste systeem gebouwd

Transition: In deze fase wordt de eerste versie van het systeem opgeleverd.

Voor meer informatie over de uitvoering van RUP zie: https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf

Voordelen en nadelen

Voordelen:

- RUP is een iteratieve manier om te ontwikkelen

- Beheersbaar en traceerbaar
- Proces is aan te passen

Nadelen:

- Zware software ontwikkelingsmethode, veel artefacten, activiteiten en rollen gedefinieerd.
- Tooling is duur
- Te veel formaliteit
- Minder klant betrokkenheid

(Runeson & Greberg, 2008) (Saiedian & Dale, 1999)

Tools:

RUP maakt gebruikt van vele tools om requirements op te stellen, processen te modelleren of het genereren van test cases. In het algemeen wordt een tool gebruikt die het gehele proces te ondersteunen, dit worden ook wel requirements management tools genoemd. Hieronder is een lijst met mogelijke tools die het proces kunnen ondersteunen:

Organisatie:	Link:	Belangrijke kenmerken:
blueprint	http://www.blueprintsys.com/	➢ Look and feel Microsoft product ➢ In cloud ➢ Simulaties maken ➢ Stakeholders worden betrokken met de tool ➢ Social conversaties ➢ Integratie Microsoft producten(zoals Visio, Excel en word
jama	http://www.jamasoftware.com/	➢ Project dashboards ➢ Uitgebreid change management ➢ Permissions ➢ Test management ➢ Social medium ➢ Controle scope
Borland	http://www.borland.com/Products/Requirements-Management/Caliber	➢ Real time impact analyse ➢ Real time Agile zorgt ervoor dat user stories overeenkomen met business requirements ➢ Requirements visualisatie
MODERN Requirements By eDevTECH	http://www.modernrequirements.com/integreat/#Features	➢ Integratie met TFS ➢ Simulaties via browser ➢ Met Simulation Designer tool (net als Axure) ➢ Look and feel Microsoft office product ➢ Integratie Microsofts

		producten(zoals Visio, Excel en word) ➢ Simpel ➢ Veel vrijheid
KOVAIR	http://www.kovair.com/alm-studio/requirements-management/	➢ Monitoring ➢ Import van Office ➢ E-mail notificatie ➢ Baseline ➢ LDAP integratie
RequirementONE make your next project better	http://www.requirementone.com/	➢ Bugs & issues ➢ Stakeholder feedback ➢ Project Planning ➢ Elke software methode ondersteund ➢ Lifecycle management
casecomplete	http://casecomplete.com/	➢ Track changes ➢ Look and feel Microsoft office ➢ Gericht op Agile
SPARX SYSTEMS	http://www.sparxsystems.com.au/products/ea/index.html	➢ Gebaseerd op UML 2.5 ➢ Archimate/BMPN ➢ Scripting ➢ Code engineering (PHP,JAVA,DELPHI,C++) ➢ Krachtige database modeling tool
	http://www.visual-paradigm.com/	➢ Requirements/wireframes/impact analyses ➢ Prototypes maken ➢ Zowel agile als waterval ➢ Verschillende modelleringstechnieken ➢ Cloud ➢ Codesource geïntegreerd ➢ Managen Project

Agile Unified Proces

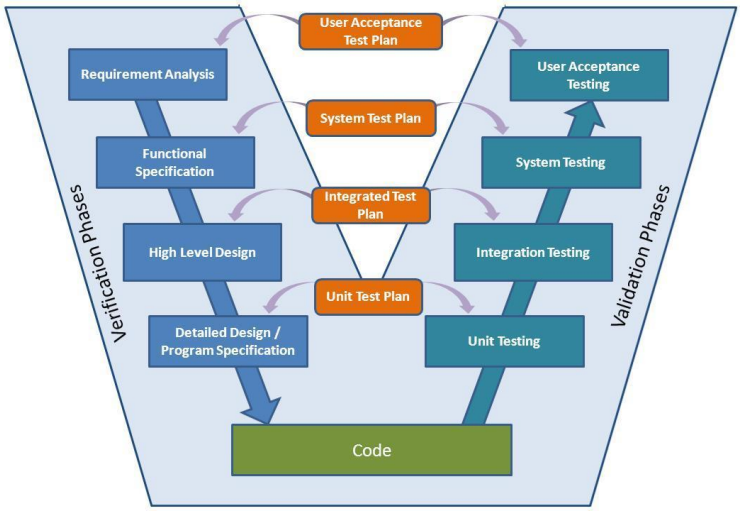
AUP is een eenvoudige versie van RUP. De methode hanteert wel dezelfde methodes, maar gebruikt meer agile technieken. Het voordeel van de methode is dat het zowel gemanaged kan worden en tegelijktijdig niet teveel artefacten vraagt van de ontwikkelaar. Het nadeel van de methode is dat het nog niet erg bekend is, waardoor er weinig ondersteuning voor is (Scott, 2006).

AUP werkt ook met andere principes ten opzichte met RUP, namelijk:

- Het personeel weet wat je aan het doen bent
- Simplistisch:
    - Het gaat om korte en krachtige documentatie
- Behendigheid:
    - De methode past zich aan op de waardes en principes van Agile Alliance.
- Focus gelegd op alleen de waardevolle activiteiten
- Tool onafhankelijk:
    - Je kan zelf bepalen welke tools voor je proces wilt gebruiken

8.5 V-model

Met het V-model wordt op een watervalachtige manier gewerkt. Je begint met het opstellen van de requirements, schermen / prototypes maken, ontwikkelen en vervolgens testen. Het V-model heeft als pluspunt vergeleken met andere watervalachtige methode dat elke fase lineair wordt getest en gevalideerd.(zie afbeelding)



De methode kan het beste gebruikt worden in kleine of middelgrote teams, waarbij de requirements volledig gedefinieerd en worden begrepen.

Tools

Voor het V-model worden geen specifieke tools gebruikt. Alleen bij de activiteiten om requirements op te stellen en schermafbeeldingen/prototypes te maken worden er requirement tools voor gebruikt.(vorige hoofdstuk beschrijft een aantal tools daarvoor)

Voordelen en nadelen

- Eenvoudige controle op planning en activiteiten
- Gemakkelijk te gebruiken
- Artefacten kunnen als testresultaat worden gebruikt
- Elke fase wordt getest

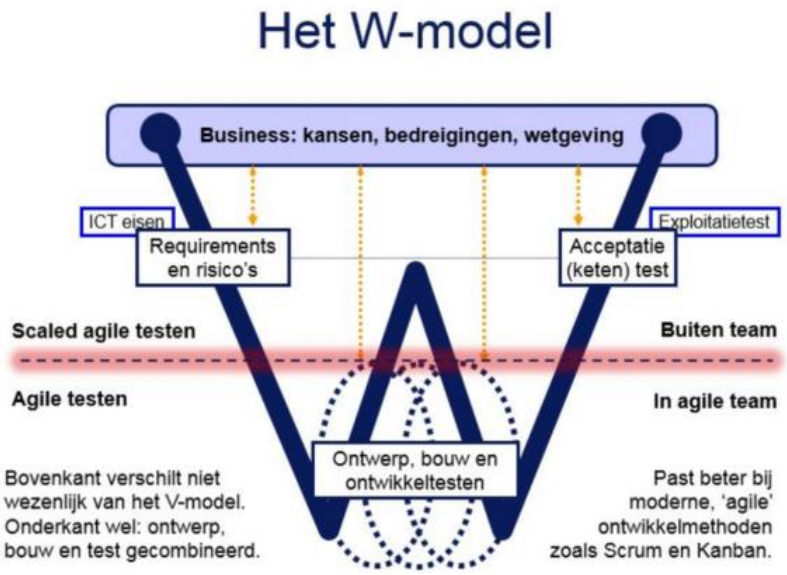
Nadelen

- Fouten in het systeem worden laat ontdekt
- Product is pas te gebruiken aan het einde van de methode
- Requirements zijn in het begin niet altijd compleet
- Weinig flexibiliteit

(Coley Consulting, 2015) (de Wille, 2014).

8.6 W-model

Veel organisaties combineren een watervalachtige methode met een agile achtige methode. Een variant hiervan is het W-model. De bovenkant van het W-model verloopt nog hetzelfde als van het V-model. Aan de onderkant wordt het interessant. Daarin wordt op een iteratieve manier ontwerpt, gebouwd en ontwikkeltesten gedaan. Aan het einde van elke iteratie/sprint wordt kort met de stakeholders naar het resultaat gekeken. Als alle iteraties zijn afgerond wordt het systeem getest met de eindgebruikers voor de oplevering. (Bouman, 2013)



De nadelen van het V-model zouden ook in dit model verholpen zijn.

Voor extra uitleg over het W-model zie: https://www.youtube.com/watch?v=StV8aqoradM&feature=youtu.be

8.7 Lean software ontwikkeling

Lean software ontwikkelings, ookwel lean thinking genoemd, is een softwareproces dat gebruikt worden samen met Agile-methodieken. De methode focust hoe je het beste waarde creëert voor de klant. Lean software ontwikkeling bestaat uit zeven principes (Scott, Strategies for Scaling Agile Software Development, 2008) (Poppendieck & Poppendieck, 2006):

Elimineer verspilling	Met verspilling wordt bedoelt, alles verwijderen wat geen klantwaarde heeft. Met verspilling wordt bedoeld: - Onduidelijke requirements - Langzame interne communicatie of processen - Onnodige code/functionaliteit - bureaucratie Om de verspilling tegen te gaan wordt een value stream mapping techniek gebruikt om het op te sporen en te verwijderen. Dit proces wordt een aantal keer herhaalt. Voor informatie over de value stream mapping zie: https://www.youtube.comtest/watch?v=nOLWp99bCsI
Bouw kwaliteit	Bij de oplevering wil je niet dat er nog veel rework wordt gedaan. Om bugs en fouten te voorkomen kan er worden gebruik gemaakt van test driven development en pair programming(in hoofdstuk 5.1 uitlegd). Bij TDD wordt eerst de test geschreven en daarna pas gecodeerd.
Stimuleer kennis	Plannen is belangrijk, maar leren is essentieel. Deze strategie wil je ook stimuleren. Op een interactieve manier werken leert de ontwikkelaar wat de werkelijke eisen en wensen van de stakeholders zijn. Dagelijkse evaluatie/feedback in het proces is ook belangrijk om als team steeds verder te groeien.
Voorkom uitstel	Begin niet met ontwikkelen van een gehele specificatielijst, maar werk iteratief.

Lever snel	Als een product sneller wordt opgeleverd bij de klant, hoe sneller feedback op terug komt. Daar een snelle oplevering blijft een het team gemotiveerd en gefocust om steeds meer waarde te leveren aan het product.
Respecteer de mens	Motiveer het team, niet controleren.
Optimaliseer het geheel	Als je een effectieve oplossing wilt, moet je het grote plaatje zien. Dit betekent dat de gehele business en processen moeten worden begrepen en de kruising tussen meerdere systemen .

De principes zorgen ervoor dat de organisatie streeft naar de volgende droombestemming:

" To keep customers from changing their minds, raise the maturity of your organization to the level where it can reliably deliver what customers want so fast that they have no time to change their minds. Focus on value, flow, and people, the rest will take care of itself" **Mary Poppendieck**

Voordelen:

- Door verspilling in het ontwikkelproces te elimineren, zorgt ervoor dat het proces wordt versneld en kosten worden bespaard in het project.
- Empowerment in het ontwikkelteam helpt bij het besluitvorming in de ontwikkeling en motiveert de teamleden.

Nadelen:

- Succes in de software ontwikkeling hangt af van hoe gedisciplineerd de teamleden zijn en hoe vooraf hun technische vaardigheden zijn.
- Als een organisatie niet een persoon met de juiste business analist heeft dan kan deze methode niet bruikbaar zijn voor hen.
- Door de flexibiliteit kan de ontwikkelaar het originele doel uit het oog verliezen.

(TatvaSoft, 2015).

8.8 Crystal

Crystal is een methode(s) die gebruik maakt van verschillende technieken en processen van andere methodes. Alleen wordt bij Crystal de focus meer gelegd op mensen in plaats van artefacten en processen. Crystal wordt door elke organisatie anders gebruikt en heeft geen houvast op structuur. De methode gaat ervan uit dat je verschillende technieken en tooling toepast om te analyseren welke onderdelen wel en welke onderdelen niet passen in de organisatie.

De Crystal methode focust zich onder meer op:

- Mensen
- Iteraties
- Community
- Skills
- Talent
- Communicatie

Cystal heeft als doel dat er geen complexe methode in de organisatie plaats vindt, maar zorgt voor minder papierwerk, overhead en bureaucratie. Echter, het is niet een erg bekende agile methode en weinig informatie over te vinden. (Alistair Cockburn, 2004)

Voordelen:

- Kan worden aangepast op de grootte van het project.
- Sterke nadruk op het testen.
- De 'menselijke' component wordt beschouwd in elk aspect van het project.

Nadelen:

- Omdat de gebruiker meer in het proces wordt betrokken is het gevaarlijk als de gebruikers niet geschikt blijkt te zijn voor het proces.
- Zware methode met veel regels en processen.

(McNiel & Zeng, 2010)

8.9 Dynamice systems development method

DSDM is een agile methode gebaseerd op rapid application development. DSDM focust zich op iteratief werken en betrokkenheid van de klant. DSDM wordt vaak gebruikt bij volledige projecten (SYSQA, 2011).

Bij DSDM worden de volgende disciplines toegepast:

- Focust op wat de business nodig heeft
- Lever op tijd
- Samenwerking
- Kwaliteit staat bovenaan
- Bouw stapsgewijs vanaf een stevige basis
- Ontwikkel iteratief
- Communiceer voortdurend en duidelijk
- Toon controle

Proces

DSDM is een cyclus dat bestaat uit vijf onderdelen:

Haalbaarheid studie:	Hierin wordt gekeken of met deze methode of het projectteam de applicatie ook werkelijk ontwikkeld kan worden gerealiseerd. Hierbij worden vaak workshops gehouden.
Business studie:	In deze fase worden de requirements van de business en gebruikers vastgelegd met behulp van interviews en workshops. DSDM gebruikt de MoScow methode om de requirements te prioriteren.
Functiebepalende cyclus:	In deze fase is het doel om een prototype te ontwikkelen. Hierbij gaat het dan vooral om de functionaliteit. Het prototype wordt vervolgens getest met de eindgebruikers.
Ontwerpen en bouwen:	De prototypes worden omgezet in werkende software.

Implementatiefase:	Hierbij wordt het systeem ingevoerd in de organisatie. Ook worden gebruikers getraind om met het systeem te werken.

Technieken

DSDM maakt voornamelijk gebruik van de volgende technieken:

- Timeboxing
- MoSCow
- Prototypes
- Testen
- Workshop
- Modelleren
- Configuratie management

Voordelen en nadelen

Voordelen:

- Gebruiker wordt gezien als de eigenaar van de oplossing
- Risico wordt geminimaliseerd door iteratief te werken
- Gebruiker wordt getraind voor implementatie
- Implementatie verloopt soepel

Nadeel:

- Omdat de gebruiker meer in het proces wordt betrokken is het gevaarlijk als de gebruikers niet geschikt voor het proces blijkt te zijn
- Zware methode met veel regels en processen

(SYSQA, 2011) (Dunlap & Clifton, 2003). (TatvaSoft, 2015).

9 Hoe kan je de eisen/wensen van de klanten beter overbrengen naar de ontwikkelaars?

Requirements verzamelen is een belangrijke activiteit in het software ontwikkelingsproces. Requirements geven een logisch inzicht wat de klant wil bereiken, of de oplossing past bij de organisatie en kan als basis worden gebruikt voor het testen.

Toch wordt er weinig aandacht besteed aan het opstellen van requirements. De oorzaken daarvan is dat de klant weinig tijd heeft om de requirements te bespreken, er geen tijd voor is vanwege deadlines of de klant ziet dit proces als tijdverspilling.

Bij het opstellen van requirements wordt vaak gebruik gemaakt van interviews en observaties. Echter, het succes van al deze activiteiten is afhankelijk van hoe goed er wordt gecommuniceerd. Het is cruciaal om de requirements te definiëren met volledige betrokkenheid van de klant (Saiedian & Dale, 1999).

Zowel XP als scrum maken gebruik van korte berichten(user stories/story cards) waarop de requirements van de gebruiker staan beschreven. Alleen deze korte berichten geven geen totaal plaatje van wat het werkelijk moet worden. De berichten zijn alleen handig als hierover gesprekken worden gevoerd.

Ook mag niet worden vergeten om de niet-functionele requirements in kaart te brengen. Dit wordt vaak in het ontwikkelingsproces vergeten (Murphy, 2011).

Om requirements beter te begrijpen, kan er gebruik worden gemaakt door afbeeldingen te maken, of zelfs prototypes. Ze zeggen altijd dat een afbeelding meer dan duizend woorden beschrijft. Dit maakt het mogelijk om beter te communiceren naar ontwikkelaars en de klant van wat het werkelijke systeem kan gaan worden. Ook kan het systeem nog worden aangepast als de betrokkenen iets anders voor ogen hadden.

Een prototype maken wordt vaak gezien als een onnodige activiteit in het softwareontwikkelingsproces. Toch heeft het maken van een prototype bewezen dat het zorgt voor:

- verminderd risico om de haalbaarheidsanalyse te bepalen.
- Identificeert de echte requirements.
- Elimineert de onnodige requirements.

Prototypes zijn fundamentele artefacten als je het vergelijkt met het opstellen van andere artefacten (Saiedian & Dale, 1999).

10 Conclusie

Elke methode heeft zijn sterke en mindere sterkere punten om een softwareproject te begeleiden. Echter, niet elke methode past bij een organisatie. Scrum en extreme programming zijn sterke methodes voor organisaties die applicaties ontwikkelen in een trekkende reis. Ze beginnen ergens, maar weten nog niet waar ze gaan eindigen. Builders & Performers ontwikkelt applicaties in een vliegende reis. In het begin worden alle requirements opgesteld en in een watervalachtig proces wordt het informatiesysteem ontwikkeld. Dit zorgt ervoor dat zij applicaties kunnen aanbieden voor een fixed price. Er zijn twee methodes die aansluiten op de huidige werkwijze en de wensen van de medewerkers (zie verslag Huidige en gewenste situatie), namelijk RUP en V-model. RUP ondersteund de meeste activiteiten hoe Builders & Performers het bedrijfsproces uitvoert. Alleen is RUP een complexe methode. Bij RUP worden veel activiteiten, artefacts en rollen uitgevoerd. RUP is in dat opzicht niet geschikt voor Builders & Performers. Het V-model sluit wel aan bij de meeste activiteiten en werkwijze die momenteel door Builders & Performers worden uitgevoerd. De watervalachtige methode ondersteunt alleen niet het agile-achtige proces waar de organisatie naar toe wil groeien. Het W-model bevat wel dit iteratieve proces. Hoewel er weinig community en literatuur te vinden is over deze methode van smarttest, bevat de hybride methode wel de kern aspecten van hoe de organisatie nu werkt en hoe de organisatie in de toekomst wil gaan werken.

Het W-model sluit aan bij de huidige processen en de manier hoe de ontwikkelaars werken en willen werken. W-model is verder een simplistische methode en gaat niet uit van complexe processen. In het W-model kan er ook gebruik maken van technieken uit de best practices methodes zoals de daily stand-up en iteraties plannen. Om de methode succesvol te integreren in de organisatie, is het fundamenteel dat het management zich ook houdt aan deze methode, zodat de ontwikkelaars de methode ook gaan adopteren in de organisatie. Het W-model gaat voor Builders & Performers bieden voor structuur en betere uitwerking van de eisen en wensen.

In het document huidige en gewenste situatie is geconstateerd dat de uitwerkingen van de eisen en wensen van de klant niet voldoende waren, dit is ook mede oorzaak van rework bij de klant. Gartner zegt hier dan ook het volgende over:

“Poor requirements definition and management is a major cause of rework and friction between business and IT.”

Bij de invoering van het W-model, moet het opstellen en managen van requirements worden verbeterd. In de literatuur wordt aangeven dat het gebruik van flow diagrammen, wireframes en prototypes kunnen helpen om de requirements voor zowel de klant als de ontwikkelaar in kaart te brengen.

11 Bibliografie

Alistair Cockburn. (2004, Juli 17). *Crystal Clear*. Opgehaald van Crystal Clear Preface: http://users.dcc.uchile.cl/~nbaloian/cc1001-03/ejercicios/crystalclearV5d.pdf

Al-Saleem, S. M., & Ullah, H. (2015). A Comparative Analysis and Evaluation of Different Agile Software Development Methologies. *International Journal of Computer Science and Network Security(VOL.15)*, 39-45.

Bouman, E. (2013). *Het W-model*. Opgehaald van smarTEST: http://www.smartest.nl/verdieping/wmodel

Coley Consulting. (2015). *Types of Testing in the V-Model*. Opgehaald van Coley Consulting: http://www.coleyconsulting.co.uk/testtype.htm

de Swart, N. (2010). *Handboek Requirements.* Delft: Eburon Business.

de Vries, P. (2013). *Een beproefde aanpak om te komen tot software architecture.* Den Haag: De Haagse Hogeschool.

de Wille, E. (2014). *The V Model*. Opgehaald van The Software Experts: http://www.the-software-experts.com/e_dta-sw-process-model-V.php

Dunlap, J., & Clifton, M. (2003, september 29). *What is DSDM?* Opgehaald van CODE PROJECT: http://www.codeproject.com/Articles/5097/What-Is-DSDM

Guay, M. (2015, September 1). *Project Management 101: The Complete Guide to Agile, Kanban, Scrum and Beyond*. Opgehaald van Zapier: https://zapier.com/blog/project-management-systems/

Heuer, J. (2015, Januari 27). *WATERFALL VS. AGILE: WHY ICEBERG USES A HYBRID APPROACH FOR ARCHER DEVELOPMENT*. Opgehaald van IceBerg: http://icebergnetworks.com/waterfall-vs-agile-why-iceberg-uses-a-hybrid-approach-for-archer-development/

Hotle, M., & Norton, D. (2010, Maart 30). *The Cultural Transition to Agile Methods: From 'Me' to 'We'* . Opgehaald van Gartner: http://www.gartner.com/document/code/174915?ref=grbody&refval=1957528

Kanban. (2016, april). *Kanban*. Opgehaald van Wikipedia: https://nl.wikipedia.org/wiki/Kanban

McNiel, S., & Zeng, L. (2010). *Crystal Method*. Opgehaald van https://docs.google.com/presentation/d/1uk4Lbi8Dvf2BfEWU7i5DCmyjtk2rFRKxRh3iBo_1b7E/edit#slide=id.p

Murphy, T. E. (2011). *Agile and the Rest of the Organization: Requirements Definition and Management.* Stamford: Gartner.

Nathan, W., & Holte, M. (2016, Januari 29). *The End of the Waterfall as We Know It*. Opgehaald van Gartner: http://www.gartner.com/document/3188962?ref=solrAll&refval=163874692&qid=e1c8f00d8e5fb3f454056c1531393612

Norton, D. (2007, Augustus 14). *Agile Essence: Scrum*. Opgehaald van Gartner: http://www.gartner.com/document/512121?ref=TypeAheadSearch&qid=19063c60023f498181f13c49a60c08ee

Norton, D. (2010, December 28). *Agile Foundation: Extreme Programming*. Opgehaald van Garnet: http://www.gartner.com/document/1508414?ref=solrAll

Poppendieck, T., & Poppendieck, M. (2006). Chapter 2: Principles. In T. Poppendieck, & M. Poppendieck, *Implementing Lean Software Development: From Concept to Cash* (pp. 19-42). Boston: Pearson Education.

Riemenscheider, C. K., Hardgrave, B. C., & Davis, F. D. (2002). Explaining Software Developer Acceptance of Methodologies: A Comparison of Five Theoretical Models. *IEEE Transactions on Software Engineering*, 1135-1145.

Runeson, P., & Greberg, P. (2008). *Extreme Programming and Rational Unified Process – Contrasts or Synonyms?* Zweden: Lund University .

Saiedian, E., & Dale, R. (1999). Requirements engineering: making the connection between the software developer and customer. In *Information and Software Technology* (pp. 419-428). Omaha: University of Nebraska.

Scott, A. (2006). *The Agile Unified Process (AUP)*. Opgehaald van Ambysoft: http://www.ambysoft.com/unifiedprocess/agileUP.html

Scott, A. (2008, Januari 21). *Strategies for Scaling Agile Software Development*. Opgehaald van IBM: https://www.ibm.com/developerworks/community/blogs/ambler/entry/lean_development_governance?lang=en

Starke, S. (2013, April 29). *A Hybrid Approach to the Product Lifecycle*. Opgehaald van Tech book: https://tech-book-store.amazon.com/post/Tx1ZW9I0O0HAQLJ/A-Hybrid-Approach-to-the-Product-Lifecycle

SYSQA. (2011, April 14). *DSDM - Dynamic Systems Development Method*. Opgehaald van Sysqa: https://www.sysqa.nl/wp-content/uploads/2012/01/Intro-DSDM-1v1-KB.pdf

TatvaSoft. (2015, april 12). *Top 12 Software Development Methodologies and Its Advantages / Disadvantages*. Opgehaald van TatvaSoft: http://www.tatvasoft.com/blog/top-12-software-development-methodologies-and-its-advantages-disadvantages/#anchor10

Verheye, G. (2013, Februari 17). *Scrum values*. Opgehaald van Capgemini: https://www.capgemini.com/blog/capping-it-off/2013/02/scrum-values#about-the-author-anchor

Wells, D. (2013, Oktober 8). *Extreme Programming: A gentle introduction*. Opgehaald van XP: http://www.extremeprogramming.org/

Wikipedia. (2015, April 28). *Softwareontwikkelingsmethode*. Opgehaald van Wikipedia: https://nl.wikipedia.org/wiki/Softwareontwikkelmethode

Wilson, N. (2012, Maart 21). *Best Practices in Transitioning to Agile: Getting Started*. Opgehaald van Gartner: http://www.gartner.com/document/1957528?ref=solrAll

Wilson, N. (2012, Maart 21). *Best Practices in Transtioning to Agile: Picking a Methodology* . Opgehaald van Gartner: http://www.gartner.com/document/code/231958?ref=grbody&refval=1957528

Bijlage A, uitslagen Enquête

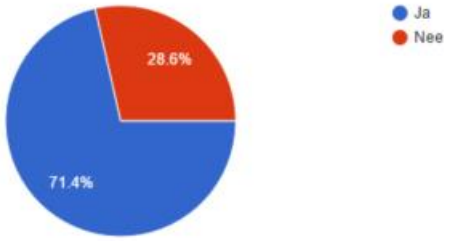
Wat is uw rol in de organisatie? (14 responses)

Sales/afstudeerder
Technisch Consultant
Stagiaire
Junior BI-Consultant
Technisch analist.
Developer
technisch analist
manager data en loyalty management
Projectmanager
Testmanager
Consultant
Team manager
Software Engineer
Manager

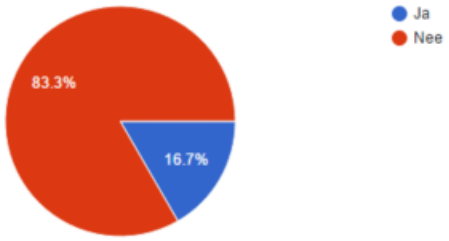
Hoe groot is uw afdeling? (14 responses)

35
35
50
50
5 mensen
80
+- 50 man
10
100
135 personen
5
15
15 medewerkers
6 mensen

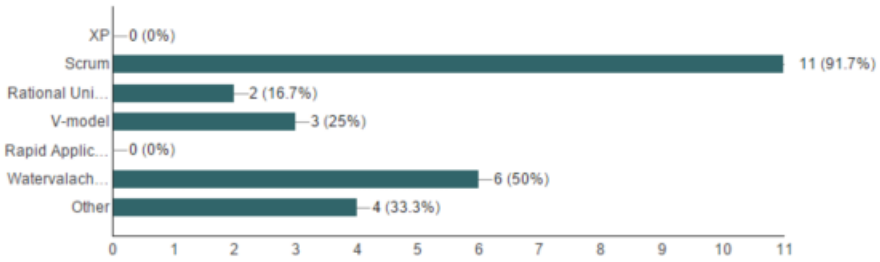
Wordt er binnen uw organisatie gebruik gemaakt van een software ontwikkelingsmethode?
(14 responses)



Zo niet, zou er behoefte zijn om gebruik te maken van een software ontwikkelingsmethode?
(6 responses)

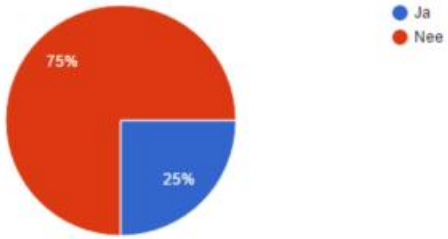


Zo ja, welke methode wordt er binnen de afdeling gehanteerd? (12 responses)



Begrijpt u de principes van de methode? (13 responses)

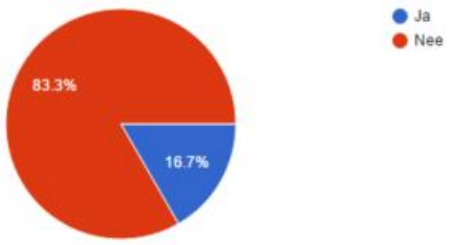
Wordt de methode ook volgens de regels nageleefd? (12 responses)



Vindt u dat de gehanteerde methode efficiënt werkt binnnen de afdeling, waarom wel, waarom niet?
(11 responses)

Ja, het biedt overzicht bij een project.
Jazeker, de Scrum methodiek wordt bijna tot in detail nageleefd en werkt zeer efficient. Niet alleen IT maar ook business werkt met Scrum.
Jawel
Scrum niet, want niet volledig doorgevoerd
nee, omdat de medewerkers hun werk niet serieus nemen.
Voor grotere projecten zeker wel. Voor veel kleinere werkzaamheden liever pragmatische oplossingen
Omslachtig
M.n. Scrum. Cultuur, mensen hebben nog niet allemaal agile mindset. Organisatiestructuur te opgehakt, wat efficiënt samenwerken complex maakt.
Op zich wel. Afhankelijk van de situatie wordt een vorm gekozen.
Nee, ze worden niet strikt gevolgd, met als reden dat we veel losse/kleine wijzigingen voor diverse klanten uitvoeren. Dus geen grote eenduidige projecten. Gevolg is wel dat opgeleverde wijzigingen nog wel eens niet voldoen aan de verwachtingen van de klant en soms kwalitatief niet voldoende zijn.
ja, omdat het efficient en prettig werkt.

Heeft u liever voorkeur voor een andere methode? (12 responses)



Welke tool(s) wordt er in de organisatie gebruikt om de methodes te structureren?
(12 responses)

Geen..
Meerdere
Jira, scrumrooms, retro's, fysieke scrumboards, de hele mikmak
Target Process
Verschillende versiebeheer tools. Verder ingerichte processen.
geen, er wordt excel, connect-people(IBM facebook achtig iets) gebruikt.
Vso en topdesk
Niet gebruikt
Jira, git, alm, icescrum
Excel, Visual Studio Online, Team Foundation Server
Geen, tenzij je TOPdesk ziet als een tool dat een development method kan ondersteunen.
Geen specifieke tools, een beetje Team Foundation Server van Microsoft

Bijlage B:

In de matrix hiernaast is geanalyseerd welke onderdelen van de methodes worden toegepast.

Rood = niet
Oranje = soms
Groen= Wel

Matrix gebruikte onderdelen methodes	scrum	kanban	rup	aup	v-model	W-model	Lean	DSDM	Watervall	XP
Proces/fases:										
Korte sprints										
JIT										
Inception										
Elaboration										
Construction										
Transition										
Haalbaarheidstudie										
Businessstudie										
Functiebepalende cyclus										
Ontwerpen en bouwen										
Implemenatiefase										
Opstellen requirements(software)										
Analyse										
Design										
Coding										
Testing										
Operations(implementatie)										
Rollen:										
Product-owner										
Ontwikkelteam(ontwikkelaars)										
Scrummaster										
Tester			?		?	?		?	?	?
Tracker										
Coach										
Communicatie:										
Daily Scrum										
Sprint Planning										
Evaluatie										
Artifacts										
User stories										
Product Backlog										
Sprint Backlog										
Acceptatiecriteria										
Burn Down										
Kanbanbord										
Business case										
Requirement engineering										
Goal-problem diagram										
Visie										
Unitesten										
Systeemontwerp										
Acceptatietest										
Business rules									?	
Datamodellen										
UML-modellen										
Principes/filosofie										
Visualiseer										
Monitoren, wijzgigen, verbeteren										
Repecteert huidige siutatie										
Accepteert incrementele verandering										
Managen requirements										
Gebruik Architectuur										
Maak prototypes										
Test het systeem										
Gebruik versie beheer			?							
Projectmanagement										
Personeel weet wat je aan het doen bent										
Simplitisch(korte documentatie)										
Focus op waardevolle activitieten										
Tool onafhankelijk										
Elimineer verspilling(simple design)										
Bouw Kwaliteit										
Stimuleer kennis										
Verkom uitstel										
Lever snel(small releases)										
Repecteer de mens(motiveer het team, niet controleren)										
Optimaliseer het geheel										
Focust op wat de business nodig heeft										
Samenwerking										
Communiceer voortdurend en duidelijk										
Toon controle										
Bouw stapgewijs vanaf een stevige basis										
Iteratief										
watervall										
Planning game										
Metaphor										
Refactoring										?
Pair Programming										
Collective Ownership										
Continous integration										
40-Hour Week										
One-sit customer										
Coding standard										?
Waardes										
Toewijding										
Focus										
Openheid										
Repect										
Lef										
Eenvoudig										
Communicatie										
Feedback										

Implementatieplan

Implementeren software ontwikkelingsmethode het W-model

Uitgebracht aan:	B&P
Uitgebracht door:	B&P
Contactpersoon B&P:	Henk Hoeksema
Author:	Ruben van Stralen
Datum:	maandag 30 mei 2016
Status:	Final, versie 1.0
Onderwerp:	Implementatie Handboek W-model

Algemene documentatie informatie

Titel	Implementatieplan
Auteur(s)	Ruben van Stralen
Datum voltooid	14-05-16
Versie	1.0

Wijzigingsoverzicht

Versie	Datum	Wat is er gewijzigd	Wie
0.1	13-04-2016	Concept versie	RUST
0.2	03-05-2016	Review versie	HEHO
0.3	03-10-2016	Review versie	RORA
1.0	14-05-2016	Final versie	RUST

Inhoud

1. Inleiding

Het implementatieplan heeft als doel om gestructureerd de processen en procedures van W-model handboek te implementeren in de organisatie. De implementatie wordt begeleidt door Ruben van Stralen(RUST), Ronny Randham(RORA) en Henk Hoeksema(HEHO). Alle stappen uit het W-model worden verschillende onderdelen/activiteiten in fases geïmplementeerd.

Met behulp van fieldresearch en de aanvulling van deskresearch is vastgesteld dat de B&P behoefte heeft aan een software ontwikkelingsmethode zodat er:

- Een methode ontstaat dat consistent het bedrijfsproces faciliterend ondersteund en optimaliseert (denk hierbij aan structuur in verantwoordelijkheid, communicatie, documentatie, planning, tooling, meetings, ect)
- Eisen en wensen gedetailleerder worden uitgewerkt.
- Kwaliteit en productiviteit van de medewerker omhoog gaat.

Het implementatieplan is tot stand gekomen door de volgende opgeleverde documentatie:

Onderzoeksrapport	Beschrijving van alle populairdere onderzocht softwareontwikkelingsmethodes en bijhorende technieken.
Rapport Huidige en Gewenste Situatie	Beschrijving van de huidige en gewenste situatie dat aanleiding is geweest tot deze verandering. Hierin is ook een lijst opgesteld met alle geprioriteerde requirements.
Handboek W-model van B&P	Kwaliteitshandboek met beschrijving van de procedures en het proces van het W-model. Het handboek wordt met dit implementatieplan in fases gerealiseerd.

De documentatie is te vinden in :

http://cloud.buildersenperformers.nl/public.php?service=files&t=951b6b44bbd6f51a478d2762935e79e

1. Achtergrond

1.1 Huidige situatie

In de huidige situatie is er geen duidelijke methode aanwezig waarmee het ontwikkelproces wordt ondersteund. Dit heeft als gevolg dat iedere medewerkers een ander beeld heeft van het proces, niet alle medewerkers bewust zijn van hun verantwoordelijkheden en de communicatielijn onduidelijk is. Daarnaast werden de requirements vanuit de BPM-er te beknopt ervaren, waardoor er vaak iets anders werd opgeleverd aan de klant. Verder is er in de organisatie ook geen duidelijke planning aanwezig van de activiteiten.

1.2 Gewenste situatie

Vanuit management wordt gevraagd om een vaste softwareontwikkeling methode te hanteren. In workshop met de medewerkers van B&P is besloten om te gaan werken via het W-model. Dit is een hybride methode tussen waterval en agile. Om het W-model consistent te gebruiken in de organisatie, is hiervoor een kwaliteitshandboek geschreven met alle procedures en afspraken. Om de procedures, processen, hulpmiddelen en verantwoordelijkheden van het handboek een plek te geven in de cultuur van de organisatie, wordt met behulp van het implementatieplan in fases het handboek geïmplementeerd.

1.3 Verandering

De verandering moet gaan zorgen dat alle medewerkers van Builders & Performers gaan werken volgens het W-model handboek ontwikkeld door Ruben van Stralen. Het W-model handboek is op te vragen in onze cloud omgeving(klant kan deze bij Henk Hoeksema opvragen). Het handboek is ontwikkeld op basis van bestaande en gewenste processen, werkwijze, technieken en is aangevuld met literatuuronderzoek.

1.4 Succesafhankelijkheid

Het succes is afhankelijk van de mate waarin de medewerkers van Builders & Performers bereid zijn mee te werken met gebruik van de procedures het W-model handboek. Voor het ontwikkelteam die gaan werken met een agile achtige methode scrum wordt geacht te veranderen naar:

- Team georiënteerd werken, en niet individueel(toewijding, openheid)
- Veranderingen geaccepteerd worden
- Doel georiënteerd werken, dus niet taak georiënteerd(focus)
- Vertrouwen(respect ,lef)

1.5 Requirements

De verandering moet zorgen dat de volgende business- en gebruikersrequirements gerealiseerd worden.

- B1: De organisatie wil haar methode/werkwijze invoeren bij B&P dat consistent het bedrijfsproces faciliterend ondersteund en optimaliseert.
- B2: De organisatie wil in staat zijn 30% meer werk op te leveren.
- B3: De organisatie wil dat de uitvoering 25% efficiënter wordt verwerkt.
- B4: De organisatie wil zorgen dat er minder frustraties plaatst vinden binnen het team.
- B5: De organisatie wil dat er 40% minder rework plaats vindt.
- B6: De organisatie wil dat de effectiviteit van de uitvoering omhoog gaat.
- B7: De opdrachtgever wil een transparante methode die gebruik maken van de methodes en technieken van agile en waterval.

Gebruikers requirement:

- G1: De ontwikkelaar wil een gedetailleerdere uitwerking van het functioneel ontwerp/specificatie.
- G2: De ontwikkelaar wil een gedetailleerdere uitwerking van de RFC.
- G3: De ontwikkelaar heeft de behoefte dat de technische uitzonderingen en functionele specificaties aan een opleverde systeem worden bijgehouden.
- G4: Het team van B&P wil een beter overzicht van de planning van de verschillende projecten.
- G6: De ontwikkelaar wil dat het inschatten van een RFC op tijd gezamenlijk verloopt.
- G7: De ontwikkelaar wil een beter communicatieplan tussen de BPM-er en de ontwikkelaars.
- G8: De BPM-er wil dat er meer bewustwording bij de ontwikkelaar komt als eindverantwoordelijke.
- G9: De BPM-er wil dat de ontwikkelaar meer zelfstandige interactie heeft met de klant wanneer het proces dit vraagt.
- G10: Het team van B&P wil heldere afspraken en structuur.

2. Onderdelen van het W-model

De volgende onderdelen van het W-model wordt met de implementatie gerealiseerd:

Onderdeel	Type:	Fase:	Beschrijving:
Daily standup	Middel	1	Elke dag wordt er met het volledige ontwikkelteam een korte staande sessie gehouden. Elke lid verteld wat die gedaan heeft, wat die gaat doen en tegen welke problemen degene aanloopt. Bij grote problemen wordt een formele vergadering gepland.
Productbacklog	Middel	1	Alle RfC's, requirements, bugs en overige items worden opgenomen in de product backlog. Deze items staan in de wachtrij om in de volgende iteratie te worden uitgevoerd. Alle items met een hoge prioriteit staan bovenaan, de items met een lagere prioriteit staan onderaan.
Scrumbord	Middel	1	Alle items die uit de product backlog voor de volgende release zijn geselecteerd, worden verdeeld in het ontwikkel team. Elke medewerkers heeft zijn eigen kleur post-its op het scrumbord om zijn items te plaatsen in verschillende statussen, namelijk de status: To-do, doing, done.
Ontwikkelteam	Organisatorisch	1	Het ontwikkelteam bestaat uit alle ontwikkelaars van B&P. Het ontwikkelteam volgt het ontwikkelproces met de methodiek scrum.
Functioneel ontwerp	Document	2	Het functioneel ontwerp is een kort en krachtig document, waarbij op een eenvoudige, consistente en duidelijke manier de functionaliteiten worden beschreven. Een functioneel ontwerp bestaat uit maximaal 2 features.
RfC forumulier	Document	2	Er is een nieuw RfC formulier gemaakt om de eisen en wensen van de klant beter in kaart te begrijpen.
Test case	Document	2	Template om het testen op te stellen van de unit-, integratie- en systeemtest
Evaluatiemeeting	Middel	2	Aan het einde van elke iteratie wordt een evaluatie meeting uitgevoerd. Het gehele team van Builders & Performers is bij de evaluatie meeting aanwezig. Het doel van een evaluatie meeting is om te kijken wat er in de iteratieperiode goed ging en wat er minder

			goed ging. Ook worden de punten uit de notulen van de vorige evaluatie meeting meegenomen. Hieruit wordt gekeken wat er in de vorige iteratie fout ging en in deze iteratie wel goed ging.
UML tool	Middel	2	De UML tool wordt gebruikt voor het maken van proces-/flowdiagrammen voor het functionele ontwerp
Wireframe tool	Middel	2	De wireframe tool wordt gebruikt voor het maken van grafische weergave van de functionaliteit(feature) voor het functionele ontwerp
Iteratie plannen	Organisatorisch	2	In deze fase wordt er met het projectteam van een klant uit de product backlog bekeken welke items(features, bugs of rfc's) in de volgende iteratieblok worden gerealiseerd. De belangrijkste items van de klant worden als eerst gekozen om te implementeren.
Overeenstemming functioneel ontwerp	Organisatorisch	2	Met de klant en de ontwikkelaars wordt het functioneel ontwerp gevalideerd. De klant beoordeeld of het functioneel ontwerp is voorzien van de functionaliteiten die hij/zij gerealiseerd wil zien. De senior ontwikkelaar beoordeeld het functioneel ontwerp of het begrijpelijk en haalbaar is binnen de afgesproken tijd.
Realiseren functioneel ontwerp	Organisatorisch	2	Bij nieuwe uitgebreide functionaliteiten wordt door de BPM-er een functioneel ontwerp opgesteld. Hierin komen alle requirements van de features in te staan met de bijhorende flowdiagrammen en/of wireframes.
Uitvoeren integratietest en tussen-acceptatietest	Organisatorisch	2	Als alle items uit het iteratieblok zijn verwerkt, wordt er door de BPM-er een integratietest uitgevoerd. Hierbij worden alle gebouwde items gezamenlijk getest op functionaliteit. Daarna wordt er door een klant een tussen-acceptatietest uitgevoerd. Bij de tussen-acceptatietest wordt niet alleen de nieuwe functionaliteiten getest, maar ook gekeken of de klant nog nieuwe eisen en wensen heeft.
Uitvoeren iteratieblok	Organisatorisch	2	In deze procedure worden de gekozen items uit het product backlog gerealiseerd door de ontwikkelaars. Bij complexe en/of nieuwe functionaliteiten wordt er door de senior ontwikkelaar eerst een technisch ontwerp opgesteld. Daarna worden de items onder de

			ontwikkelaar verdeeld om vervolgens te bouwen.
Wijziging aan bestaand systeem	Organisatorisch	2	Bij deze procedure wordt beschreven hoe de RfC's en bugs van bestaande klanten in kaart wordt gebracht. De RfC's en bugs worden vervolgens in de product backlog geplaatst. Hierbij wordt ook gebruikt gemaakt van het nieuwe RfC formulier.
Acceptatiedocument	Document	3	De acceptatietest wordt uitgevoerd door de klant opgesteld door de BPM-er. De acceptatiedocument test de geïmplementeerde functionaliteit(requirements)
Business Case	Document	3	Voor elke nieuwe klant wordt een business case opgesteld. In de business case worden alle doelstellingen, planning, kosten en baten, risico's, business requirements en features beschreven door de projectmanager. Aan de hand van een business case geeft de klant een go or no go.
Requirementlijst	Document	3	Voor elke informatiesysteem van een klant wordt door de BPM-er een requirementlijst bijgehouden. In de requirementlijst worden alle business requirements, features en gedetailleerde requirements gedefinieerd.
Technische Ontwerp	Document	3	Bij een technisch ontwerp wordt bepaald **HOE** we de software gaan maken. In het technisch ontwerp wordt gekozen welke technieken er worden gebruikt, hoe de data wordt opgeslagen en hoe de ingevoerde gegevens worden verwerkt.
BPM-er	Organisatorisch	3	Brengt alle gedetailleerde requirements van de klant in kaart. Dit verwerkt de BPM-er in een requirementlijst en functioneel ontwerp. Tevens is BPM-er betrokken bij de integratie- en acceptatietesten.
Buitenteam	Organisatorisch	3	Het buitenteam bestaat uit projectmanager, bpm-ers en opdrachtgevers.
Definiëren gedetailleerde requirements	Organisatorisch	3	Bij akkoord op de business case wordt in de volgende procedure door de BPM-er de gedetailleerde requirements in kaart gebracht. De requirements kunnen achterhaald worden door de klant te interviewen, observeren, workshop houden of eventueel te observeren.
Documentatiemanagement	Organisatorisch	3	Per project wordt een map aangemaakt om alle gemaakte documenten zoals de business case, functioneel ontwerp en technisch ontwerp in te

			plaatsen.
Project initiatie	Organisatorisch	3	In de eerste procedure wordt er voor een nieuwe klant een business case opgesteld. Het doel van de procedure is om tot een akkoord te komen van de business case. De business case beschrijft o.a. de achterhaalde business requirements en features die de klant gerealiseerd wil hebben van het informatiesysteem. Ook zorgt de projectmanager ervoor dat in deze procedure alle stakeholders van de klant in kaart wordt gebracht om in de volgende fase de gedetailleerde requirements te achterhalen.
Project Manager	Organisatorisch	3	De project manager bespreekt met de opdrachtgever af wat het systeem moet opleveren. Aan de hand van de gekregen informatie stelt de project manager de business case op.
Uitvoeren acceptatietest	Organisatorisch	3	Hier worden alle features door de klant getest. Als de klant tevreden is over het opgeleverde informatiesysteem, dan wordt het project afgesloten.
Uitvoeren systeemtest	Organisatorisch	3	Als alle items uit het product backlog zijn verwerkt van de klant, dan wordt er door de BPM-er/senior ontwikkelaar een systeemtest uitgevoerd van het gehele systeem. Hierbij worden niet alleen alle functionaliteiten getest, maar ook de performance van het informatiesysteem.
Projectmanagementtool	Middel	4	Projectmanagement tool heeft het voordeel verschillende projecten alle activiteiten digitaal te managen. Ook geeft een project management tool een duidelijk overzicht van alle taken. Met een projectmanagement tool kan een medewerker eenvoudig zien welke werkzaamheden die nog moet verrichten, welke nog open staan en welke al gedaan zijn.
Requirement management tool	Middel	4	Primaire doel van een requirement tool is om alle requirements en bijhorende objecten(zoals flow diagrammen, wireframes, prototypes) op te stellen en te managen.

3. Fasering

Het W-model wordt in vijf fases geïmplementeerd in de organisatie. Elke fase bevat nieuwe/gewijzigde procedures, processen, hulpmiddelen, ect. **Elke fase duurt ongeveer 3 maanden**. Daarna wordt er met de betrokkenen geëvalueerd over de afgeronde fase.

!Indien er na de afronding van een fase onderdelen van het Handboek worden toegevoegd, verwijderd of vernieuwd, dan dient het handboek hierop te worden aangepast.

3.1 Fase 1(Apr-Aug): Scrumtechnieken gebruiken

De basis van het W-model voor het ontwikkelteam is scrum. Voor implementatie dienen volgende onderdelen te worden ingericht:

- Productbacklog
- Scrumbord
- Daily Standup

De scrumtechnieken worden eerst uitgelegd voor het gehele team voordat de stappen worden gevolgd.

	Product Backlog	
Stap:	Activiteit:	Georganiseerd door:
1.	Ontwikkel productbacklog	RUST
2.	Uitleg productbacklog aan management	RUST
3.	Uitleg productbacklog aan het ontwikkelteam	RUST
4.	Faciliteren van materialen voor productbacklog	RUST/RORA
5.	Starten met registratie van RFC's en bugs in product backlog	STPL
6.	Monitoren of alles wordt bijgehouden	RUST

	Scrumbord	
Stap:	Activiteit:	Georganiseerd door:
1.	Ontwikkel scrumbord	RUST
2.	Uitleg scrumbord aan management	RUST
3.	Uitleg scrumbord aan ontwikkelteam	RUST
4.	Faciliteren van materialen voor scrumbord	RUST/RORA
5.	Monitoren of alles actief wordt bijgehouden en verwerkt in het scrumbord	STPL/RUST
6.	COULD: Aanschaf groter scrumbord en post-its magneten	RORA

	Daily Stand ups	
Stap:	Activiteit:	Georganiseerd door:
1.	Uitleg DS aan management	RUST
2.	Uitleg DS aan ontwikkelteam	RUST
3.	Monitoren of DS dagelijks wordt georganiseerd	RUST

3.1.1 Risico's

Risico	Tegenmaatregel	Kans	Impact
DS wordt top-down ingezet.	Uitleggen aan management dat de ontwikkeling bottom-up moet plaats vinden.	5	4
Productbacklog wordt niet actief bijgehouden.	Evalueer met ontwikkelteam waarom het productbacklog niet actief wordt bijgehouden.	4	4
Scrumbord wordt niet actief bijgehouden.	Evalueer met het ontwikkelteam waarom het scrumbord niet actief wordt bijgehouden.	3	4
Post-its wordt niet duidelijk geformuleerd.	Bij de DS bespreken wat de post-it inhoud en dit vervolgens bewerken.	3	2
Wordt nog te individueel gewerkt en niet team georiënteerd.	Gezamenlijk bespreken in DS wat ieder gaat doen i.p.v. dat iemand management wat ieder gaat doen.	5	5
Scrumbord/productbacklog wordt door het management bijgehouden.	Uitleggen aan management dat de ontwikkeling bottom-up moet plaats vinden.	4	5
Niet gehele ontwikkelteam wordt actief betrokken.	Bespreek met senior ontwikkelaar waarom de sommige ontwikkelaars niet actief betrokken worden.	3	4

3.1.2 Evaluatie

Aan het einde van de fase wordt er met het ontwikkelteam geëvalueerd , georganiseerd door RUST. Het doel van de evaluatie is om te achterhalen bij de ontwikkelaars of de scrumtechnieken prettig worden ervaren. Dit kan worden gedaan met behulp van een scorekaart gecombineerd met een interview.

3.2 Fase 2(Sep-Nov): Uitvoering iteratieblok van RfC's

In fase 2 wordt er met het ontwikkelteam en de BPM-er een iteratieblok gerealiseerd met behulp van het W-model. Het iteratieblok bevat RfC's van één bestaande klant verzameld in de productbacklog. Om te testen of de uitvoering van de procedures zorgvuldig verlopen, worden de iteratieblokken eerst in een zeer korte periode uitgevoerd en langzamerhand in een langere periode van 3 tot 4 weken.

1ste iteratieblok: 1 week (2x)

2de iteratieblok: 2 weken(2x)

3de iteratieblok: 3 tot 4 weken(∞)

De volgende onderdelen worden in deze fase geïmplementeerd:

- Procedure 3: Iteratieplanning
- Procedure 4: Realiseren functioneel Ontwerp
  - Functioneel Ontwerp
  - UML tool
  - Wireframe tool
- Procedure 5: Overeenstemming functioneel ontwerp
- Procedure 6: Uitvoeren iteratieblok
  - Technisch ontwerp
  - Test case
- Procedure 7: Uitvoeren integratietest en Tussen-acceptatietest
- Procedure 10: Wijziging aan bestaand systeem
  - RfC formulier

Templates zijn te vinden in:

http://cloud.buildersenperformers.nl/index.php/apps/files?dir=/Materiaal%20W-model/Templates

Procedures W-model		
Stap:	Activiteit:	Georganiseerd door:
1.	Delen W-model handboek met gehele team	RUST
2.	Lezen van de procedures 3 t/m 10	RUST
3.	Presenteer procedures 3 t/m 10 - Leg ook de verantwoordelijkheid per rol uit	RUST
4.	Monitor of de procedures grotendeels worden uitgevoerd. (ook qua verantwoordelijkheid)	RUST

	Procedures 10 en 3	
Stap:	Activiteit:	Georganiseerd door:
1.	Uitleg nieuwe RfC formulier aan gehele team.	RUST
2.	RfC formulier in Eventum implementeren	RORA
3.	Uitleg nieuwe RfC forumlier aan betrokken klant in deze fase	RUST/HEHO
4.	Verzamel RfC's/bugs van één bestaande klant in de productbacklog(genoeg om in een iteratieblok te plannen)	STPL/RORA
5.	Organiseer een iteratieplanning met ontwikkelteam - Schat hierbij de RfC's/bugs gezamenlijk op ontwikkeltijd(sharp) - Kies met het gehele team RfC's/bugs voor de volgende iteratie	STPL
6.	Monitor of na de iteratieplanning het scrum/productbacklog wordt bijgewerkt.	

	Procedure 4: Realiseren functioneel ontwerp	
Stap:	Activiteit:	Georganiseerd door:
1.	Uitleg FO template met voorbeeld uitwerking FO en bijhorende objecten aan BPM-er.	RUST
2.	Installeer wireframe tool http://pencil.evolus.vn/	RUST
3.	Uitleg wireframe tool	RUST
4.	Installeer UML tool http://www.umlet.com/	RUST
5.	Uitleg UML tool	RUST
6.	Uitleg documentmanagement aan gehele team	RUST
7.	Ondersteun BPM-er bij uitwerking FO	RUST
8.	Monitor of de procedure wordt uitgevoerd.	RUST

	Procedure 5: Overeenstemming functioneel ontwerp	
Stap:	Activiteit:	Georganiseerd door:
1.	Uitleg functioneel ontwerp aan klant	RUST
2.	Uitleg functioneel ontwerp aan ontwikkelteam	RUST
3.	Monitor of de procedure wordt uitgevoerd.	RUST

	Procedure 6: Uitvoeren iteratieblok	
Stap:	Activiteit:	Georganiseerd door:
1.	Uitleg technisch ontwerp aan ontwikkelteam.	SPTL
2.	Uitleg evaluatiemeeting iteratieblok	RUST
3.	Uitleg test case template voor unittest aan ontwikkelaars	RUST
4.	Monitor of de procedure wordt uitgevoerd.	RUST
5.	Monitoren of alles actief wordt bijgehouden en verwerkt in het	RUST

scrumbord

	Procedure 7: Uitvoeren integratietest en tussen-acceptatietest	
Stap:	Activiteit:	Georganiseerd door:
1.	Uitleg integratietest en tussen-acceptatietest aan BPM-er (ook uitleg template test cases en acceptatieformulier)	RUST
2.	Ondersteun BPM-er bij uitvoering testen	RUST
3.	Monitor of de procedure wordt uitgevoerd.	RUST

3.2.1 Risico's

Risico	Tegenmaatregel	Kans	Impact
scrumtechnieken worden niet meer actief bijgehouden.	Bespreek met senior ontwikkelaar waarom de scrumtechnieken niet meer worden toegepast.	4	4
Procedures van het handboek worden niet goed doorgenomen door het team.	Organiseer tijdens de presentatie een quiz over de procedures, de winnaar wordt beloond met een fles eigen gebrouwen whisky van ronny. Een quiz kan eenvoudig gemaakt worden op: https://getkahoot.com/	4	5
Procedures worden amper gevolgd.	Plan een meeting met het gehele team om te bespreken waarom de procedures amper worden gevolgd.	4	5
Templates worden niet ingevuld of gebruikt.	De templates kunnen niet goed begrepen of gecommuniceerd zijn, ondersteun ze dan hierbij.	2	4
Klant is niet actief bij bespreking FO en/of tussen-acceptatie.	Communiceer naar de klant belang van actieve betrokkenheid.	3	5
De toegewezen verantwoordelijke neemt de rol niet op zich.	Bespreek individueel met diegene waarom hij de verantwoordelijkheid niet op zich neemt.	4	5

3.2.2 Evaluatie

Aan het einde van deze fase wordt ook geëvalueerd met de betrokken klant. De contactpersoon Henk Hoeksema bespreekt met de klant hoe hij/zij het functioneel ontwerp en tussen-acceptatie ervaarde. De informatie van de klant wordt ook gebruikt voor de evaluatie met het team.

Met het B&P team wordt geëvalueerd over de gerealiseerde releases met het gebruik van het W-model(een betrouwbare klant hierbij betrekken is aan te raden). Er wordt in de vergaderruimte drie grote velden geplakt. Iedere medewerker krijgt een eigen kleur post-its. De velden hebben de volgende titels:

- Wat ging er goed
- Wat kon er beter
- Tips om het W-model voor B&P nog beter te maken

Iedere medewerker mag post-its plakken op alle drie de velden. Zorg ervoor dat de medewerkers ongeveer 10/15 minuten ruimte krijgen hiervoor. Daarna wordt er gezamenlijk gesproken over de post-its. Maak tijdens de evaluatie ook notities en vooral foto's van de velden!

3.3 Fase 3(Dec-Feb): Nieuw project volgens W-model

In deze fase wordt een nieuw project (dus geen bestaande klant) gerealiseerd met behulp van het W-model. Doel van deze fase is dat het W-model wordt uitgevoerd met alle stakeholders. In deze fase worden alle procedures, processen, eindverantwoordelijke, documentatie en principes uitgevoerd. Ten opzichte van de vorige fase worden de volgende procedures toegevoegd aan het proces:

- Procedure 1: Project Initiatie
  - Business case
- Procedure 2: Definiëren gedetailleerde requirements
  - Requirementlijst
- Procedure 8: Systeemtest uitvoeren
- Procedure 9: Applicatietest uitvoeren

	Nieuw project volgens W-model	
Stap:	Activiteit:	Georganiseerd door:
1.	Zoeken van een nieuwe klant om het gehele W-model uit te voeren. - Project moet niet langer duren dan 3 maanden	RUST
2.	Uitleggen overige procedures aan gehele team(leg daarbij ook de verantwoordelijkheid uit)	RUST
3.	Uitleggen W-model aan klant	RUST
4.	Uitleggen template business case en bijhorende objecten aan management en project manager	RUST
5.	Overleggen met management voor behoefte aannemen BPM-specialist (liefst met kennis IIBA)	RORA
6.	Uitleggen projectmanager en BPM-er over opstellen verschillende type requirements met de requirementlijst	RUST
7.	Monitoren of alle procedures zorgvuldig worden uitgevoerd, dus ook de scrumtechnieken, documentatie en toegewezen verantwoordelijke	RUST
8.	Uitleggen en ondersteunen bij uitvoering systeemtest en acceptatietest	RUST
9.	Meten resultaat of de business requirements behaald zijn(zie hoofdstuk 3.3.2)	RIST

3.3.1 Risico's(zie ook risico's vorige fase)

Risico	Tegenmaatregel	Kans	Impact
Klant begrijpt het W-model niet	Extra informeren of voorzien van handboek	2	3
Nieuw project is te klein	Ander project kiezen om met deze fase uit te voeren.	3	4
Requirementlijst wordt niet bijgehouden	-	2	3

3.3.2 Evaluatie

Fase 3 is een essentiële fase van het implementatieplan. Nadat fase 3 met een nieuwe project is gerealiseerd wordt er door RUST, HEHO en RORA geanalyseerd of de opgestelde business requirements(van het afstudeerproject, niet van de klant) zijn behaald. Er wordt gekeken of:

- Of er minder rework plaatst vond, in staat was meer werk op te leveren, efficiënter gewerkt wordt.
- Minder frustraties waren binnen het team.(heb met elke ontwikkelaar een individueel gesprek hierover).
- Effectiviteit van de uitvoering omhoog is gegaan.

Het gaat hierbij om de business requirements B1 t/m B6. Zie hoofdstuk 4 van rapport Huidige en gewenste situatie.

Een aantal business requirements kunnen op de volgende manier gemeten worden:

Business Requirement	Hoe meten?
B2: ***"De organisatie wil in staat zijn 30% meer werk op te leveren binnen dezelfde tijd".***	Dit kan worden gemeten door het aantal afgeronde RfC's in Eventum per ontwikkelaar. Hierbij wordt het aantal gelijkwaardige afgeronde RfC's vergeleken met het aantal gelijkwaardige afgerond RfC's 9 maanden terug.
B3: ***"De organisatie wil dat de uitvoering 25% efficiënter wordt verwerkt".***	Dit kan worden gemeten worden door de hoeveelheid uren rework na oplevering op te tellen of het aantal gewerkte uren meten dat niet in rekening is gebracht.
B4: ***"De organisatie wil zorgen dat er minder frustraties plaats vinden binnen het team".***	Dit kan eventueel gemeten worden door het aantal uren ziekteverzuim op te tellen. Verder kan met een gezamenlijk overleg of observatie worden gekeken of de sfeer is verbeterd.

B5: ***"De organisatie wil dat er 40% minder rework plaats vindt".***	Dit kan worden gemeten door de het aantal uren rework uit Eventum op te tellen van een iteratie 9 maanden terug en het aantal uren rework op te tellen na de uitvoering van de iteratie in deze fase.
B6: ***"De organisatie wil dat de effectiviteit van de uitvoering omhoog gaat."***	Dit kan worden gemeten door op te tellen hoe vaak er in één keer goed wordt opgeleverd.

Als de business requirements niet zijn behaald, dan wordt met het gehele team geëvalueerd of het W-model nog past in de organisatie. In een overleg wordt er besproken of de organisatie wil doorgaan met het handhaven van het W-model.

3.4 Fase 4(Apr-Jun): Tooling

Om het W-model te ondersteunen met tooling kan er gebruikt worden gemaakt van een projectmanagenttool en een requirementmanagement tool.

Project management tool		
Stap:	Activiteit:	Georganiseerd door:
1.	Overleggen met ontwikkelteam of het fysieke scrumbord wel of niet vervangen moet worden door een projectmanagementtool	RUST
2.	Analyseren naar de verschillende projectmanagement tool(met kosten) Reeds voldaan(zie onderzoeksrapport, adviseer JIRA)	RUST
3.	Rapporteren naar management over de verschillende projectmanagement tools	RUST
4.	Kiezen met het gehele team voor geschikte project management tool	RORA
5.	Inrichten projectmanagement tool	RUST
6.	Presenteer projectmanagement tool voor gehele team(eventueel ook voor klant)	RUST
7.	Ondersteun gehele team met gebruik van de projectmanagement tool	RUST
8.	Monitoren actief gebruik van het projectmanagement tool	RUST

Stap:	**Activiteit:**	**Georganiseerd door:**
1.	Overleggen met management of de requirementslijst moet worden vervangen door een requirement management tool	RUST
2.	Analyseren naar de verschillende projectmanagement tool(met kosten) Reeds voldaan(zie onderzoeksrapport, adviseer JAMA)	RUST
3.	Rapporteren naar management over de verschillende requirement tools	RUST
4.	Kiezen met management voor geschikte requirement management tool	RORA
5.	Inrichten requirement management tool* - Instellen templates - Mappenstructuur - Rollen definities - Proces	RUST
6.	Presenteer requirement tool voor gehele team(eventueel ook voor klant)	RUST
7.	Ondersteun BPM-er/project manager met gebruik van de projectmanagement tool	RUST

8. Wijzig het handboek door gebruik requirement tool.	
9. Monitoren actief gebruik van het Requirement management tool.	RUST

***** Afhankelijk van de gekozen requirement tool.

Risico	Tegenmaatregel	Kans	Impact
Tools worden niet actief gebruikt.	Onderzoek wat de oorzaak daarvan is.(met behulp interview)	3	3
Tools worden niet begrepen	Handleidingen maken Extra presentatie/uitleg	4	4
Investeringskosten zijn te hoog	Baten scherp stellen, anders terug gaan naar oude situatie	4	3

3.4.1 Evaluatie

Evalueer met gehele team of de geïmplementeerde tool(s) effectief is geweest binnen de organisatie. Anders moet er een rollback worden gedaan naar de oude situatie.

4. Kwaliteitswaarborging

De implementatie wordt uitgevoerd in vijf fases, met ieder een duur van drie maanden. Om te meten of de technieken, procedures en andere onderdelen uit het W-model aansluiten op de gewenste bedrijfsvoering, wordt er na elke fase geëvalueerd. De evaluatie wordt georganiseerd door Ronny. Afhankelijk van het onderdeel, wordt er besloten of de evaluatie individueel of met het gehele team wordt gehouden.

Aan het einde van fase 3 wordt gemeten of de opgestelde business requirement worden behaald. De analyse bepaald of het W-model effectief is binnen de gewenste bedrijfsvoering.

Bij wijzigingen op de uitvoeringen van het W-model, wordt er door Ronny of Henk het gehanteerde handboek gewijzigd. Dit geldt ook voor wijziging in de templates.

Bijlage A: Evaluatie fase 1:

Evaluatie STPL

Hoe vond je het eerst gaan en hoe vind je het nu gaan?

(op een schaal van 1 tot 10)

12 weken terug: 5,5

Nu: 7

In het begin was het nooit duidelijk wat er gedaan moest worden, met behulp van de scrumtechnieken is dit een stuk duidelijker geworden. Hierdoor heb ik meer rust gekregen voor mijn eigen werk en zijn de gesprekken die we hebben een stuk nuttiger dan eerst. Vroeger hadden we gesprekken wat er moest gebeuren, nu hebben we gesprekken hoe het moet gebeuren. En dat is een hele vooruitgang.

Hoe heb je het functioneel ontwerp ervaren?

Cijfer: 8

Het functioneel ontwerp heb ik positief ervaren. Het zorgde ervoor dat de requirements van de klant beter begrepen werden door de ontwikkelaars, mede dankzij de extra informatie per requirement en het gebruik maken van de wireframes. Daarnaast zorgt het functioneel ontwerp ervoor dat de specificatie is opgesplitst in kleine brokjes en goed aan elkaar gelinkt zijn. Dit zorgt ervoor dat we nauwkeuriger en stapsgewijze kunnen ontwikkelen/testen.

De wireframes kunnen qua traceerbaarheid iets duidelijk, door ze bij de feature zelf neer te zetten.

Hoe heb je het scrumbord ervaren?

Cijfer: 8

Het was voor ons nog even zoeken hoe we het bord moesten inrichten. De eerste versie werkte naar mijn mening nog niet heel goed. De huidige versie hoe we het nu hebben, dus een kleur post-it per medewerker en een template hoe ze de post-it moeten invullen, maakt het overzicht voor mij wat er gedaan wordt per medewerker een stuk duidelijker.

Een punt voor verbetering is dat iedere medewerker moet letten om zijn eigen post-its up to date te houden. Dit wordt nu nog niet altijd gedaan. Ik denk hoe langer we het scrumbord gebruiken, hoe meer het een plek krijgt in de werkwijze van een medewerker om het bord up to date te houden.

Hoe heb je de daily stand ups ervaren?

Cijfer: 6

De reden dat ik de daily stand up nog niet zo'n hoog cijfer geef komt omdat de ze naar mijn mening nog te lang duren. Het doel van een daily standup is om een korte meeting te houden van rond de 15 min wat er wordt gedaan. Nu duren de daily standups rond de 30 minuten. Er wordt nu eigenlijk nog te veel besproken over wat er niet gehaald is. En dat is niet de bedoeling, dat kunnen we beter bespreken in een evaluatie. Ik denk ook dat het voor de ontwikkelaars nog even wennen is. Ook zou het helpen als we met ze alle op dezelfde tijd binnen zijn. Nu komt het nog wel is voor dat iemand 3 uur later dan mij binnenkomt. Dan heeft het voor mij al bijna geen nut om een daily standup te houden, aangezien ikzelf toch al halverwege me werkdag zit.

Zal door de verandering de werkdruk afnemen?

Nog niet, ik denk dat het nog een stuk gestroomlijnder kan. Maar dat had ik ook nog niet verwacht in 10 weken, ik denk dat de verandering nog even tijd nog heeft om uiteindelijk een plek te krijgen in de organisatie. Daarna zou ook mijn werkdruk een stuk afnemen.

Zou de kwaliteit hierdoor toenemen?

Dat wel, wat ik al eerder vertelde, nu hebben we gesprekken hoe het moet gebeuren i.p.v. wat er moet gebeuren. Er is nu zinnig overleg, hierdoor gaat de kwaliteit zeker omhoog.

Laatste woorden

Gewoon volhouden!

Evaluatie HAHO

Hoe vond je het eerst gaan en hoe vind je het nu gaan?

(op een schaal van 1 tot 10)

12 weken terug: 5

Nu: 7

In de eerste weken was er amper overleg en wat er gedaan moest worden qua activiteiten was erg onoverzichtelijk. Door de introductie van het scrumbord is het ontwikkelproces voor mij een stuk overzichtelijker geworden wat er gedaan moet worden. Iedereen weet nu van elkaar met welke activiteiten wij bezig zijn, dat was hiervoor niet. Door het scrumbord wordt ook de releases voor ons een stuk overzichtelijker. Daarnaast hebben we nu ook veel vaker overleg in het ontwikkelproces.

De volgende stap waar we ons in moeten verbeteren in de uitwerking van de tickets in Eventum.

Hoe heb je het functioneel ontwerp ervaren?

Cijfer: 6

Ik heb het FO dat jij hebt gemaakt nog een 6 gegeven. Qua structuur is het voor ons een stuk duidelijker. Alleen inhoudelijk zaten een aantal fouten, waardoor er rework heeft plaatsgevonden. Ook kwam het FO nog niet overeen met de RfC. Exta informatie in de requirements (bijv. de formulebereking FMV-waarde in excel). Dus eigenlijk wens ik nog meer HOE het gedaan moet worden, i.p.v. wat er gedaan moet worden.

Hoe heb je het scrumbord(product backlog) ervaren?

Cijfer: 7

Het scrumbord helpt ons ontwikkelproces enorm. We hebben nu een stuk overzichtelijker beeld wat er gedaan moet worden. De verschillende kleuren post-its helpt daar ook een enorme bijdrage bij. Ik denk de volgende stap die we moeten zetten is een groter whitebord te nemen met daarbij van die magneet post-its. Dit zou het scrumbord een stuk efficiënter maken.

Daarnaast stond ook het product backlog. Dit kunnen we goed gebruiken voor directe issues, alleen alle andere openstaande tickets is het verstandig om de product backlog digitaal bij te houden. Dus in Eventum. Anders verlies je het overzicht wat er allemaal in de toekomst nog op het programma staat.

Hoe heb je de daily stand ups ervaren?

Cijfer: 7

De daily standups zorgen ervoor dat wij als team regelmatig overleggen wat er gedaan moet worden. Als de daily stand up wordt vergeten, dan wordt vaak door iemand in ons team hieraan herinnert om te doen. Tijdens de daily stand up bespreken we niet alleen wat er gedaan moet worden, maar ook hoe. Dit zorgt ervoor dat we na de bespreking goed kunnen doorwerken en dat we elkaar minder van het werk afhouden.

Soms wordt er nu nog wel een vergeten na de daily standup om de post-its up to daten. Dit kunnen we in ons team nog wel een stuk verbeteren.

Zal door de verandering de werkdruk afnemen?

De werkdruk is iets afgenomen, maar voornamelijk de irritaties die ik vroeger had zijn door de scrumtechnieken sterk afgenomen. Met behulp van het scrumbord en daily standup weet ik wat er gedaan moet worden, hierdoor ervaar ik veel minder stress en werk ik hierdoor een stuk prettiger.

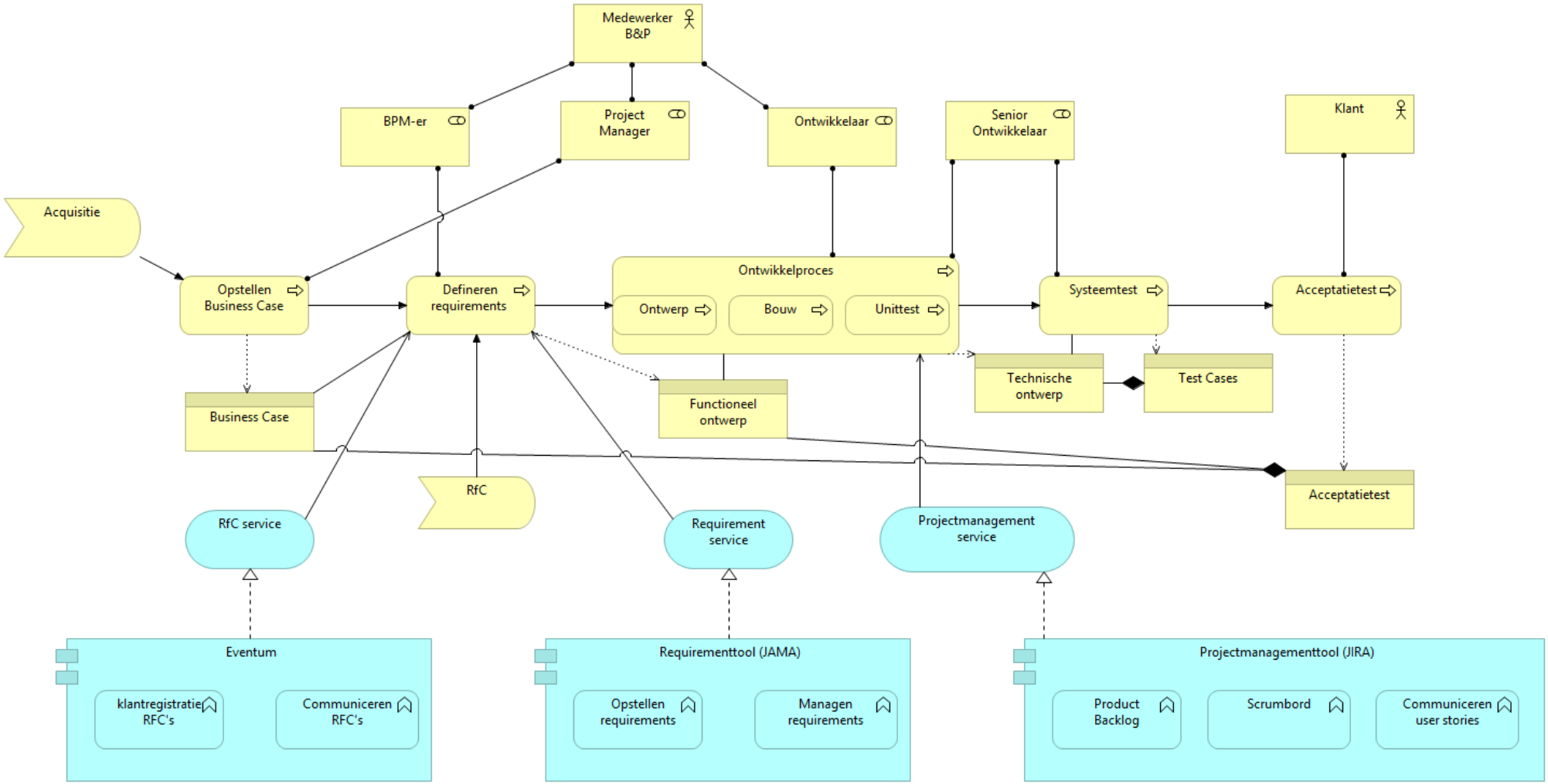
Zou de kwaliteit hierdoor toenemen?

Jazeker. Vooral nu we met het scrumbord ook het testen echte taak zien, zorgt ervoor dat de kwaliteit omhoog gaat.

Tips

Zoals ik al eerder zij, een groter backlog. Ook zou ik de tickets in product backlog van Eventum beter uitgewerkt zien. Zodat we tijdens een iteratieplanning begrijpen wat er gedaan moet worden van de opgestelde tickets.

Bijlage B: Architectuur na implementatie



! Als het proces/tooling is gewijzigd tijdens de implementatie, dan moet de architectuur worden aangepast!

Link naar bijlage D
Hanboek: https://www.dropbox.com/s/b4xsh4s2xzqqq9n/BIJLAGE%20D_Handboek_W-model.pdf?dl=0