

Afstudeerverslag

Optimaliseren van het beheer van ontwerpdocumentatie

Versie 1.0

Auteur: B.H.M. van Heugten

Opleiding: Fontys Hogeschool ICT
Bedrijfskundige Informatica

Datum: 7 juni 2012

Optimaliseren van het beheer van ontwerpdocumentatie

Auteur:	Benny van Heugten
Studentnummer:	2120498
Jaar van uitgave:	2012
Opdrachtgever:	Guy Bruens (g.bruens@infoland.nl) Manager product development
Bedrijf:	Infoland De Run 6801 5504 DW Veldhoven 040 848 58 68
Opleiding:	Bedrijfskundige Informatica (voltijd) Fontys Hogeschool ICT Rachelsmolen 1 5600 AH Eindhoven 0877 877 299
1ste afstudeerdocent:	Jolanda Liégeois
2de afstudeerdocent:	Rob van Stratum
Afstudeerperiode:	februari t/m juni 2012
Akkoord bedrijfsbegeleider:	Datum: _____ Paraaf: _____

Voorwoord

Voor u ligt de scriptie die is geschreven naar aanleiding van het onderzoek 'Optimaliseren van het beheer van ontwerpdocumentatie' dat ik heb uitgevoerd bij Infoland. Dit afstudeeronderzoek is onderdeel van de opleiding Bedrijfskundige Informatica aan de Fontys Hogeschool ICT te Eindhoven. De afstudeeropdracht is uitgevoerd in de maanden februari t/m juni 2012.

De afstudeerperiode is een traject geweest waarbij onderzoek is gedaan naar de huidige en gewenste situatie bij Infoland om uiteindelijk een goed advies te kunnen formuleren. Ik heb het onderzoek met veel motivatie en gedrevenheid uitgevoerd. Hierdoor heb ik met plezier gewerkt aan dit onderzoek en ben ik dankzij Infoland een ervaring rijker.

Vanuit deze positie wil ik graag een aantal mensen bedanken voor de ondersteuning en begeleiding die zij mij hebben geboden tijdens het afstudeertraject. In de eerste plaats wil ik Infoland bedanken voor het aanbieden van een afstudeeropdracht. Guy Bruens en Hilbert Leijen wil ik bedanken voor de begeleiding vanuit Infoland. Rob van Stratum dank ik zeer hartelijk voor de geweldige begeleiding vanuit school. Naast mijn begeleiders wil ik ook de collega's van Infoland bedanken voor de hulp en medewerking die ik gekregen heb. Als laatste wil ik mijn ouders en broertje bedanken voor de steun gedurende de gehele afstudeerperiode.

Benny van Heugten

Veldhoven, juni 2012



Inhoudsopgave

SAMENVATTING.....	5
SUMMARY	6
1 INLEIDING	7
2 HET BEDRIJF	8
2.1 INLEIDING	8
2.2 ORGANISATIE.....	8
2.3 INFOLAND SOFTWARE (SOFTWARE SOLUTIONS).....	9
3 DE OPDRACHT.....	10
4 DE AANPAK.....	11
4.1 INLEIDING	11
4.2 METHODES	11
4.3 PROJECT PLANNING.....	14
5 INITIEEL ONDERZOEK	15
5.1 INLEIDING	15
5.2 VOORONDERZOEK.....	15
5.3 NADER ONDERZOEK	17
6 DIEPTEONDERZOEK.....	21
6.1 INLEIDING	21
6.2 HET ONDERZOEK	21
6.3 FUNCTIONEEL ONTWERP.....	23
6.4 PRODUCTCATEGORIEËN	26
6.5 PAKKETSELECTIE	27
6.6 CONCLUSIE	37
7 OPLOSSINGSPLAN	38
7.1 INLEIDING	38
7.2 KNELPUNTEN	38
7.3 SOFTWARE OPLOSSING	39
7.4 OPLOSSINGEN.....	41
8 CONCLUSIES EN AANBEVELINGEN.....	44
EVALUATIE	45
LITERATUURLIJST	46
BIJLAGEN.....	
A. PROJECT INITIATION DOCUMENT.....	
B. NADER ONDERZOEK	
C. SOFTWARE LIJST	
D. KNELPUNTEN (OPLOSSINGSPLAN)	
E. SOFTWARE OPLOSSING (OPLOSSINGSPLAN).....	

Samenvatting

Infoland is een bedrijf dat zich heeft gespecialiseerd in het ontwikkelen van software op het gebied van kwaliteitsmanagement en risicobeheersing. Op Software Solutions afdeling werken software architecten, software engineers en test engineers intensief samen aan nieuwe releases van de software, in cycli die zich zo'n 3x per jaar herhalen. Binnen het software ontwikkelproces speelt het functioneel ontwerp voor de software een grote rol. Het functioneel ontwerp wordt zeer gedetailleerd opgesteld door de software architecten. Dit functioneel ontwerp is vervolgens de leidraad waarmee de ontwikkelaars en testers aan de slag gaan.

De omvang en complexiteit van het functioneel ontwerp is door de jaren heen gegroeid. En nu kost het veel werk en tijd om het functioneel ontwerp te onderhouden en te beheren. De opdrachtgever heeft gevraagd om onderzoek te doen naar de knelpunten ten aanzien van het beheer en onderhoud van het functioneel ontwerp, om zo advies uit te kunnen brengen hoe het proces geoptimaliseerd kan worden. Als aanvulling is ook gevraagd om onderzoek te doen of er software is voor het beheren en fabriceren van het functioneel ontwerp.

Door middel van interviews af te leggen met betrokkenen zijn de knelpunten in kaart gebracht. Met die informatie is er diepgaand onderzoek gedaan naar de mogelijke oplossingen en voornamelijk in de richting van software. Daarom is er een pakketselectie uitgevoerd in de categorie die uit onderzoek na boven kwam namelijk Rapid Prototyping tools (Mockups/Wireframes) en Graphical user interface builder tools. Deze pakketselectie heeft een softwarepakket opgeleverd genaamd Axure.

Tijdens het onderzoek bleek wel dat de softwarematige oplossing niet alle knelpunten zou wegnemen. En omdat een implementatie van het softwarepakket als Axure te lang zou duren, worden de volgende punten aanbevolen om het beheer en onderhoud te optimaliseren: Onderzoek doen naar mogelijke procesmatige veranderingen in het gehele software ontwikkelproces; Onderzoek doen om de bestaande software te optimaliseren of te overwegen om een geheel nieuwe software toepassing te ontwikkelen; Onderzoek doen naar het verbeteren van het ToDo formulier, die ingevuld worden wanneer er onjuistheden ontdekt zijn in het functioneel ontwerp, zodat belangrijke gegevens bij het invullen niet over het hoofd worden gezien; Onderzoek doen naar het opstellen van een checklist ter ondersteuning bij het nakijken van het functioneel ontwerp; De ontwikkelingen van het softwarepakket Axure in de gaten houden mocht het softwarepakket in de toekomst wel gaan voldoen aan de wensen van Infoland.

Summary

Infoland is a company that specializes in developing software in the field of quality management and risk management. In a cycle that repeats 3 times a year, the Software Solutions department that includes software architects, software engineers and test engineers are working intensively together for the new software release. The functional design document has a major role within the software development process. It is made with great detail by the software architects. This functional design document serves as a major guidance in the work of the software engineers and the test engineers.

Over the years, the functional design document has evolved in size and complexity. Causing a burden of maintaining and managing the functional design document, because it costs more time and effort. Infoland has asked to investigate bottlenecks and give advice on how to optimize the functional design in regards to maintaining and managing it. In addition to that questions they also asked to research whether there is software for managing and manufacturing a functional design document.

By doing an interview with experts that are involved in the process, it was possible to find out what the bottlenecks are. With this information it is possible to investigate further into the possible solutions, and then mainly in the direction of a software solution. Therefore a software package selection was used to find out what the best software solution is. To come up with the best solution the direction of what software category had to be investigated during the research, and it came up with the category Rapid Prototyping tools (Mock-ups / Wireframes) and graphical user interface builder tools. The software package selection came up with the best solution named Axure.

During the research it became clear that the software solution won't completely resolve all problems. Besides that the implementation of Axure would take a long time. But with the knowledge of this research the following can be recommended to optimize the maintainability and manageability of the functional design: Research further into possible process-related changes over the entire software development process; Research into options of optimizing the current software or developing a complete new software application; Doing research in improving the ToDo form, which is entered when inaccuracies are discovered in the functional design after handing it over, so that important information will not be overlooked while filling it in; Research into creating a checklist to assist the reviewing process of the functional design; It is also recommend to monitor Axure development, because it maybe will meet Infoland needs in the future.

1 Inleiding

Infoland is een bedrijf dat zich specialiseert in het ontwikkelen van software op het gebied van kwaliteitsmanagement en risicobeheersing. Om die software te kunnen ontwikkelen stellen de architecten een zeer gedetailleerd functioneel ontwerp op. Dit doen ze door gebruik te maken van MS Word en MS Visio. Het functioneel ontwerp is vervolgens de leidraad waarmee de ontwikkelaars en testers aan de slag gaan. De complexiteit en omvang van de functionele ontwerpen is door de jaren heen gegroeid. Het kost veel werk om het functioneel ontwerp te beheren. Het komt bijvoorbeeld voor dat dezelfde wijzigingen moet worden doorgevoerd in verschillende versies van het functioneel ontwerp.

Daarom is dit project gestart met als doel om advies uit te brengen naar Infoland hoe zij het beheer en onderhoud van het functioneel ontwerp kunnen verbeteren. Om er achter te komen waar de knelpunten liggen zijn er interviews afgelegd met betrokkenen. Aan de hand van die interviews is er diepgaande onderzoek uitgevoerd. Het diepgaande onderzoek heeft geresulteerd in het uitvoeren van een pakketselectie waarbij één pakket het beste uit de selectie kwam. Dit softwarepakket is verder onderzocht om te kijken of die wel degelijk bij Infoland past. Het projectresultaat heeft ook geleid tot andere mogelijke oplossingen ten aanzien van de knelpunten.

Dit document zal een beschrijving geven van de resultaten en verantwoording afleggen van de werkzaamheden. In hoofdstuk 2 van dit document wordt het afstudeerbedrijf Infoland beschreven. Hoofdstuk 3 geeft verdere toelichting op de opdracht. In hoofdstuk 4 wordt de aanpak en de gebruikte methodes beschreven. Hoofdstuk 5 beschrijft het initieel onderzoek waarin het voor- en nader onderzoek is uitgevoerd. In hoofdstuk 6 wordt het diepteonderzoek besproken waarin een stuk literatuuronderzoek is uitgevoerd en de pakketselectie. Hoofdstuk 7 beschrijft het oplossingsplan in het plan worden de mogelijke oplossingen beschreven en wordt de software oplossing gereflecteerd op Infoland. Als laatste hoofdstuk 8 waar de conclusies en aanbevelingen van dit onderzoek zijn te vinden.

2 Het bedrijf

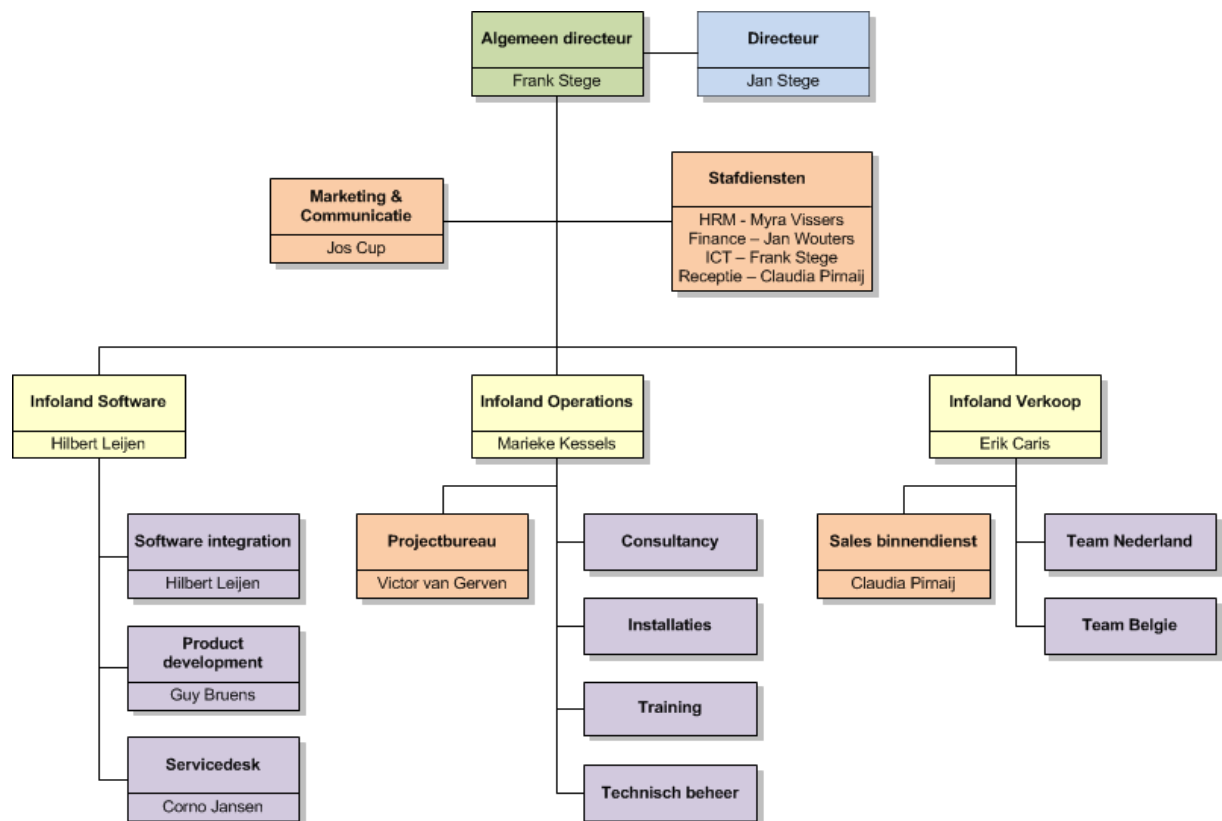
2.1 Inleiding

In dit hoofdstuk wordt het bedrijf weergegeven. Eerst wordt de organisatie als geheel besproken en vervolgens de Software Solution afdeling waar de afstudeerstage heeft plaatsgevonden.

2.2 Organisatie

Infoland is een relatief jonge en innovatieve organisatie die zich in haar meer dan 10 jarig bestaan heeft ontwikkeld tot een vooraanstaande softwareproducent voor kleine, middelgrote en grote organisaties in de zorg, het onderwijs en het bedrijfsleven. De oplossingen die Infoland biedt, zijn geïntegreerde totaaloplossingen op het gebied van kwaliteits- en veiligheidsmanagement, web content management en e-learning. Infoland biedt complete dienstverlening in de vorm van software en consultancy op een persoonlijke en professionele manier. Het hoofdkantoor van Infoland is gevestigd in Veldhoven. Daarnaast heeft Infoland een vestiging in Houten en Geel (België). Infoland is een sterk marktgerichte organisatie met korte lijnen. Daarom kan Infoland snel en flexibel insprijnen op de wensen en eisen van hun klanten. In totaal werken er ruim 75 goed opgeleide medewerkers bij Infoland.

Hieronder Figuur 1 het organogram van de gehele organisatie.



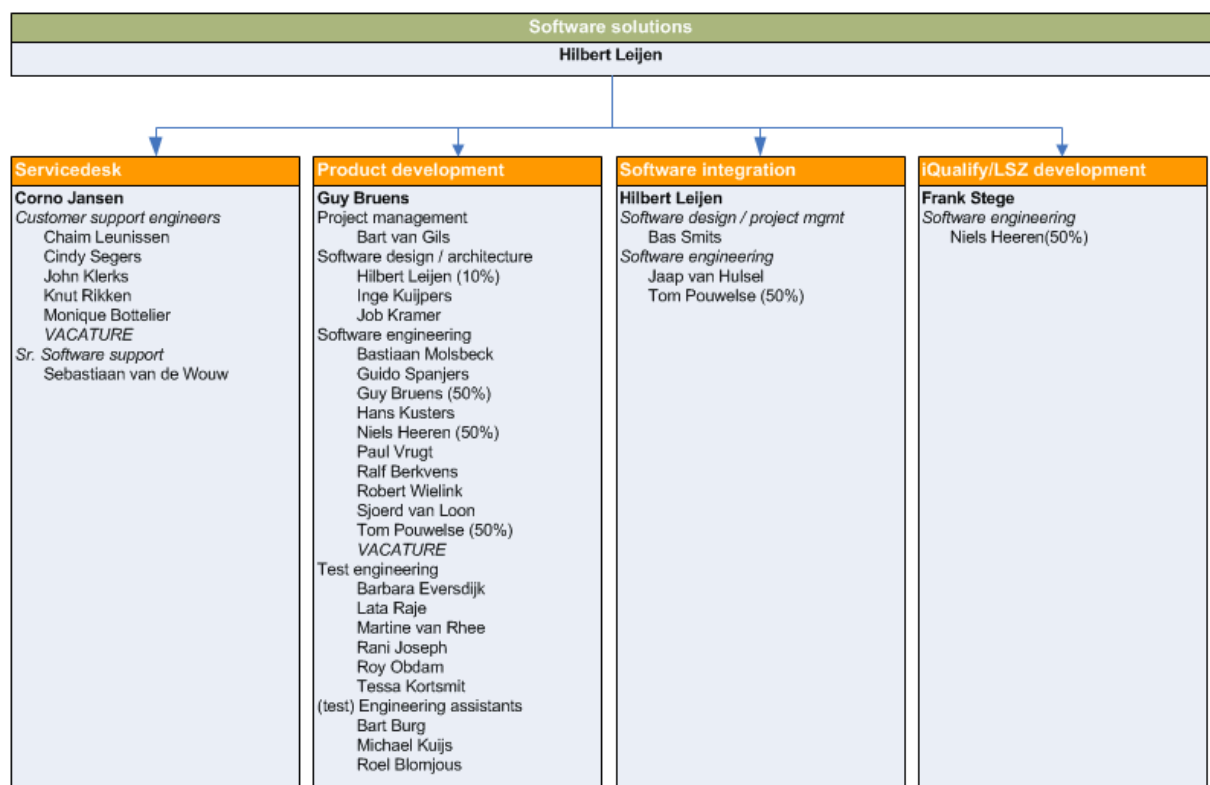
Figuur 1 - Organogram Infoland

Figuur 2 - Hoofdkantoor Veldhoven



2.3 Infoland Software (Software Solutions)

Binnen de afdeling Software Solutions bestaan er vier teams: het maatwerkteam (Software Intergration), het product development team, de servicedesk en Leerstation Zorg. Het maatwerkteam draagt zorg voor het aanpassen van de software die Infoland levert aan haar klanten. Hierbij kan gedacht worden aan het aanpassen van het kleurenschema en het toevoegen van logo's. Het product development team maakt de basisapplicaties. Het team bestaat uit software architecten, software engineers en test engineers. De servicedesk onderhoudt het contact met klanten en is het aanspreekpunt voor het melden van problemen. Zie Figuur 3 - Organogram Software Solutions.



Figuur 3 - Organogram Software Solutions

3 De opdracht

Het team Product Development zorgt binnen Infoland voor het bouwen en het onderhouden van de standaard software. Op deze afdeling werken software architecten, software engineers en test engineers intensief samen aan nieuwe releases van de software, in cycli die zich zo'n 3x per jaar herhalen. Binnen het software ontwikkelproces speelt het functioneel ontwerp voor de software een grote rol. De architecten stellen het functioneel ontwerp op, de software engineers gebruiken het om de software te realiseren en de testers testen aan de hand van het functioneel ontwerp of het klopt wat de software engineers gemaakt hebben. Op het ogenblik kost het veel werk om het functioneel ontwerp bij de tijd te houden vandaar de naam van het project "Optimaliseren van het beheer van ontwerpdocumentatie".

Het doel van dit project is om advies uit te brengen naar Infoland hoe zij het beheer en onderhoud van het ontwerpdocument kunnen verbeteren. En voor zover de beschikbare tijd dit toelaat een korte implementatiefase waarin de nieuwe methode en/of tool in de praktijk wordt gedemonstreerd in de vorm van een proefproject c.q. een proof-of-concept.

Vragen die aan het einde van het project beantwoordt moeten zijn:

- Welke problemen/knelpunten doen zich voor in het huidige software ontwikkelproces ten aanzien van het ontwerpdocument (functioneel ontwerp) en hoe op te lossen?
- Hoe beheert en onderhoudt men een ontwerpdocument (functioneel ontwerp) op de juiste manier?
- Is er software voor het beheren/fabriceren van een ontwerpdocument (functioneel ontwerp), hoe werkt deze, welk is geschikt voor Infoland en hoe kan die dan het beste ingericht worden voor Infoland?

4 De aanpak

4.1 Inleiding

In dit hoofdstuk wordt de globale aanpak van dit afstudeerproject beschreven. Zo wordt duidelijk welke methodes er gebruikt zijn en wordt de planning besproken. De originele aanpak staat beschreven in het Project Initiation Document (PID) bijlage A.

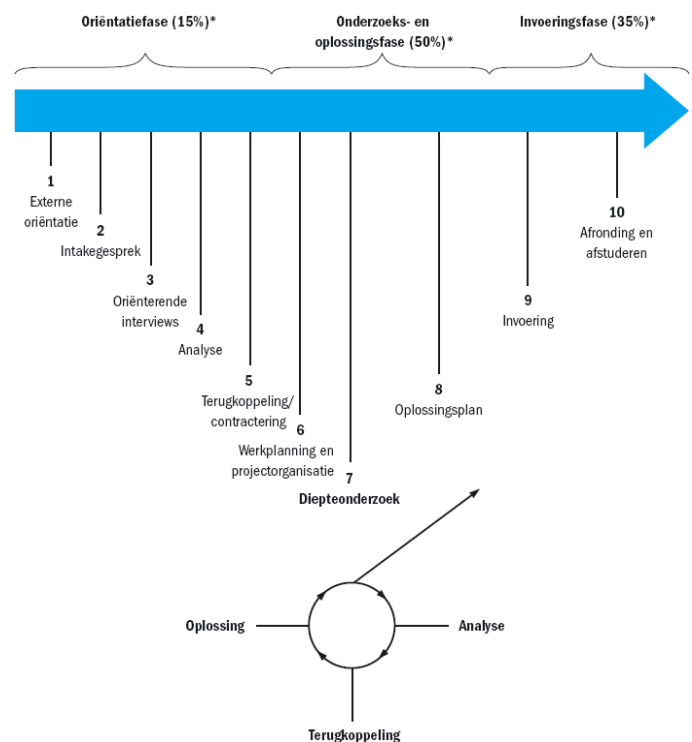
4.2 Methodes

De gebruikte methodes in dit project zijn Tienstappenplan (TSP), Prince2, V2-methode en Indora. Ook zijn er aspecten gebruikt als Quality Function Deployment (QFD) methodiek, Gewogen Factor Score (GFS) methode en kwalitatieve onderzoeksmethodes. De methodes zullen hier kort behandeld worden.

4.2.1 Tienstappenplan (TSP)

Tijdens het afstuderen is er gebruik gemaakt van het Tienstappenplan. De methode is ontwikkeld door Piet Kempen en Jimme Keizer. Door het tienstappenplan heeft de afstudeerder meer grip op het afstudeertraject en biedt het effectiever begeleiding. Men zag gelijkenis tussen een professioneel adviestraject en een afstudeertraject. Uitgaande van een gebruikelijk adviestraject bedachten zij voor afstudeerders het Tienstappenplan om de kwaliteit van het afstudeertraject te borgen.

Vanuit de opleiding wordt ook dit tienstappenplan aangehouden. Uit Figuur 4 valt op te maken dat het afstudeertraject bestaat uit 10 stappen en dat deze stappen onder zijn verdeeld in 3 verschillende fases, namelijk de: oriëntatiefase, onderzoeksfase en invoeringsfase.



* Aanbevolen tijdsverdeling van het totale project in fasen van het TSP

Figuur 4 - Tienstappenplan

Met behulp van deze fases wordt de afstudeerder in staat gesteld een zeker afstudeertraject te volgen. Door gebruik te maken van het tienstappenplan kan de afstudeerder aantonen dat hij competent kan afstuderen. Deze stappen reflecteren terug in de Project Initiation Document (PID), initieel onderzoek, diepteonderzoek en het oplossingsplan.

4.2.2 Prince2

Het idee achter Prince2 is eigenlijk alleen gebruikt voor het schrijven van de Project Initiation Document (PID). Dit document is te vinden in de bijlage A. Wel vond er elke keer controle plaats ten aanzien van de planning. Dit liep vrijwel equivalent aan de planning zoals TSP het hanteert. De reden voor Prince2 niet verder te hanteren is dat dit project niet proportioneel is tegen over de methode. Het schrijven van een PID wordt ook vanuit de opleiding aangehouden.

4.2.3 V2 Verbetermethode

De V2 verbetermethode is een bundel dat deel uit maakt van het documentatiemateriaal dat gebruikt wordt bij diverse onderdelen van de studie Bedrijfskundige Informatica. De bundel bevat een aantal methoden en technieken die gebruikt worden om systematisch te onderzoeken, te adviseren en te veranderen. De V2-methode is rond 1990 ontstaan bij de (post-HBO) cursus "Kwaliteitskunde". Cursisten, afkomstig uit het bedrijfsleven, hebben de V2-methode in de praktijk bij "verbeteren-opdrachten" toegepast. Naar aanleiding van het commentaar van die cursisten is de methode diverse keren aangepast, zodat uiteindelijk een praktisch toepasbare methode is ontstaan. De V2-methode is voor een deel gebaseerd op algemeen toegankelijke literatuur en voor een deel gebaseerd op eigen ervaringen van de betrokkenen.

De V2-methode kent vergelijkbaar fases als het TSP namelijk: Diagnose fase, Therapie fase en Implementatie fase. In dit project is V2-methode voornamelijk gebruikt in het initieel onderzoek fase waarbij de ist (huidige situatie) en soll (gewenste situatie) in kaart gebracht moesten worden.

De keuze voor de V2-methode en niet andere verbetermethode is dat V2-methode het beste van alles bundelt. Buiten het feit dat men al bekend is met gebruik van deze methode in de opleiding.

4.2.4 Indora

Indora is de pakketselectiemethode die gebruikt is in het diepteonderzoek. Het is opgesteld door een adviesbureau Indora Informatisering. De methode biedt een leidraad die past bij dit project, omdat de stappen zoals Indora ze beschrijft overeenkomen met de stappen die gebruikelijk zijn voor V2-methode en TSP. Ook levert Indora checklists waardoor men aan het denken wordt gezet. Verdere toelichting om deze methode en de keuze daartoe is te vinden in hoofdstuk 6.5 waar de pakketselectie wordt uitgelegd.

4.2.5 Quality Function Deployment (QFD)

Quality Function Deployment is een methode om klantenwensen en marktvragen te vertalen in ontwerpeisen. Het resultaat van deze methode is een gegevensmatrix. De kritische analyse die wordt toegepast binnen dit project is aan Quality Function Deployment ontleend. De koppeling van knelpunten uit de ist-situatie aan oplossingsmogelijkheden uit de soll-situatie levert een kritische analyse matrix op. Ook wordt dezelfde methode gebruikt in het oplossingsplan om de tool te ramen op Infoland.

4.2.6 Gewogen Factor Score (GFS)

De gewogen factor score methode is een compensatorische methode en wordt veel gebruikt om de meeste geschikte offerte te kiezen. Op basis van gedefinieerde criteria worden de offerten beoordeeld door middel van een scoren. De criteria worden gewogen door middel van een schaalscore. Vervolgens wordt de schaalscore vermenigvuldigd met de scoren van de offerten. Ieder bij elkaar opgeteld levert een totaalscore op, de offerte met de hoogste score komt het beste uit de test. De gewogen factor score methode wordt in dit project gebruikt bij de pakketselectie om via deze methode het meest geschikte softwarepakket te selecteren.

4.2.7 Methode van gegevens verzamelen

Er wordt gebruik gemaakt van kwalitatieve onderzoeksmethodes. Hieronder vallen het interviewen van betrokkenen, het observeren, inhoud analyse/documentanalyse voor de interne documenten en de gevonden informatie op het internet. Ook het discussiëren en vergaderen valt informatie mee in te winnen.

4.3 Project planning

De planning is opgesteld aan de hand van de vooraf bekende data. Het TSP was een goed raamwerk om de planning op te stellen. De proof-of-concept is wel in de planning verwerkt alleen bevat het geen zekerheid dat dat het eindresultaat is. Als de voorgaande fases meer tijd in beslag nemen kan er besloten worden om de proof-of-concept niet uit te voeren. Zie Tabel 1 – Project planning.

Item	Datum van	Datum tot	Aantal dagen
Oriëntatiefase	6-feb	17-feb	10
Externe oriëntatie	6-feb	7-feb	
Intakegesprek	8-feb	10-feb	
Interviews	13-feb	17-feb	
Vooronderzoek	15-feb	22-feb	
PID	17-feb	22-feb	
Initieel onderzoek	20-feb	24-feb	5
Overleg	20-feb	-	
Nader onderzoek	20-feb	24-feb	
Diepteonderzoek	27-feb	6-apr	30
Samenvatting Nader onderzoek	2-apr	6-apr	
Literatuuronderzoek	27-feb	6-apr	
Oplossingen	26-mrt	6-apr	
Advies	26-mrt	6-apr	
Tussentijdse presentatie	9-apr	13-apr	5
Oplossing plan	16-apr	11-mei	20
Research	16-apr	27-apr	
Verbeteringen & efficiency zaken	30-apr	11-mei	
Proof-of-concept	14-mei	1-jun	15
Scriptie inleveren	4-jun	7-jun	
Eind presentatie + eindrapport enz.	4-jun	20-jun	13
Totaal dagen:			98

Tabel 1 – Project planning

De planning is doorberekend tot 20 juni dit is de laatste dag voor de geplande afstudeerzittingen.

5 Initieel onderzoek

5.1 Inleiding

In dit hoofdstuk worden de onderdelen van het initieel onderzoek besproken. Zo heeft het initieel onderzoek geleid tot een vooronderzoek en nader onderzoek. Het nader onderzoek eindigt met een conclusie.

5.2 Vooronderzoek

Het vooronderzoek is uitgevoerd in twee stappen namelijk het houden van de oriënterende interviews en het uitvoeren van een analyse. Beide gecombineerd heeft geleid tot een vooronderzoeksrapport in dit rapport wordt aangegeven wat de knelpunten zijn en hoe de huidige situatie bij Infoland in elkaar steekt.

5.2.1 Oriënterende interviews

Tijdens de oriëntatiefase zijn er interviews gehouden met de betrokkenen om zo een juist beeld te krijgen van de huidige en gewenste situatie binnen Infoland. Het interview heeft een half-gestandaardiseerde opzet en hiermee wordt bedoeld dat er van te voren onderwerpen zijn bepaald en vragen zijn opgesteld door de interviewer. Tijdens het interview heeft de respondent de mogelijkheid om alle informatie aan te dragen die hij/zij van belang acht. Bij stilvallen van het interview kan de interviewer één onderwerp aandragen waar nog niet over gesproken is. Ook zal de interviewer controleren of alle van te voren bedachte vragen tijdens het interview worden beantwoord, zo niet zullen deze vragen alsnog gesteld worden. Informatie zal na het interview verwerkt worden in één verslag door de interviewer.

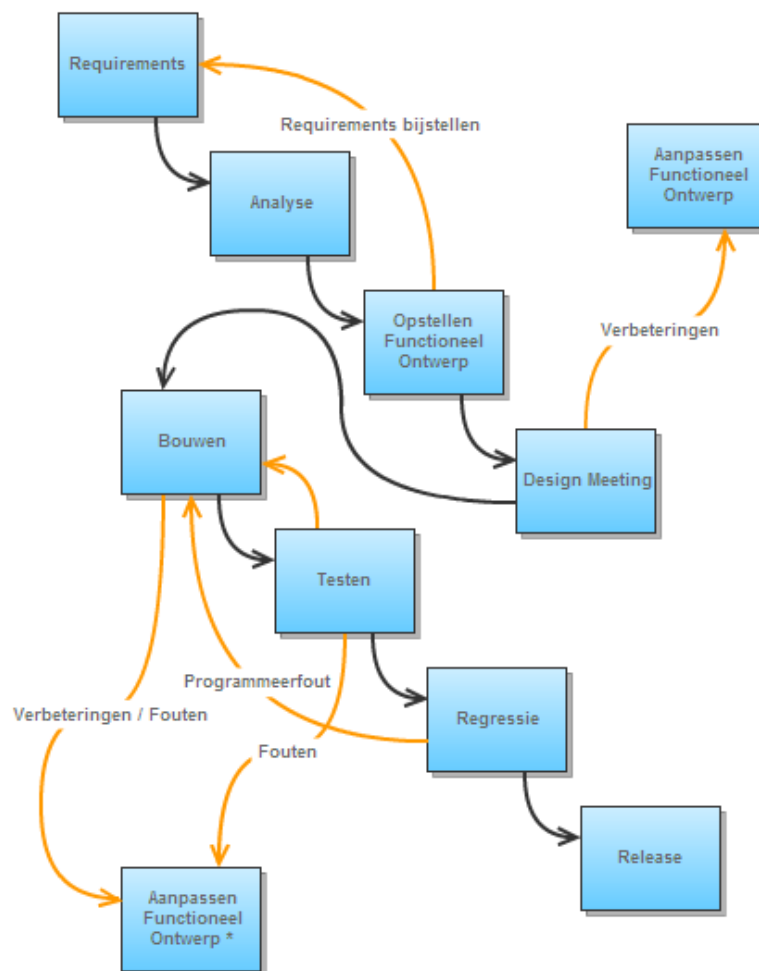
De betrokkenen die geïnterviewd zijn voor dit project is één (lead) software engineer, test specialist, twee software architecten en de manager van de afdeling. Er is expliciet gekozen om personen te interviewen in verschillende delen van het proces. Hierdoor kan er een beter beeld gecreëerd worden van de knelpunten/wensen op het gebied van het ontwerpdocument (functioneel ontwerp).

Uit de interviews zijn de volgende knelpunten na boven gekomen:

1. Er zijn verschillende versies van het functioneel ontwerp;
2. Wijzigingen in bestaande functioneel ontwerp zijn niet direct zichtbaar;
3. Functioneel ontwerp bestand is heel groot;
4. Het kost veel tijd/werk om delen van het functioneel ontwerp aan te passen;
5. Het functioneel ontwerp niet compleet of er ontbreken verwijzingen;
6. Geen indicatie wanneer ToDo's afgehandeld worden.

5.2.2 Analyse

Om de knelpunten zoals ze in het vorige hoofdstuk zijn opgenomen te kunnen beschrijven is er een oorzaak en gevolganalyse uitgevoerd. Voordat dat wordt toegelicht eerst meer duidelijkheid over de huidige processituatie in hoofdlijnen. Er is gekozen om het software ontwikkelproces in kaart te brengen, dit is het primaire proces van de afdeling product development. In Figuur 5 hieronder is het proces zichtbaar waarbij de zwarte pijlen de standaard watervalmethode illustreren en de oranje geven de aanpassingen aan die door Infoland zijn toegevoegd aan het proces.



Figuur 5 - Software ontwikkelproces Infoland

De omschrijving van iedere stap is te vinden in bijlage B nader onderzoeksrapport. Het software ontwikkelproces wordt in een cyclus 3x per jaar uitgevoerd. De eerste 4 fases in Figuur 5 worden uitgevoerd door de architecten die 3x per jaar achter elkaar herhalen. Dit houdt in dat de software engineers aan het ontwikkelen zijn en de architecten zijn al bezig met het bedenken/uitwerken van de volgende release.

Om tot een correcte beschrijving te komen van de knelpunten, die na voren zijn gekomen tijdens de interviews, is er gekozen de oorzaken en gevolgen te beschrijven. Dit diende als hulpmiddel het juiste probleem/knelpunt te beschrijven. Zo kan een bepaald knelpunt meerdere oorzaken en gevolgen hebben. Ook biedt deze uitwerking een goed beeld van wat er allemaal opgelost moet worden om een knelpunt geheel weg te nemen.

Een voorbeeld dat speelt bij Infoland; is dat architecten niet tegelijk aan het zelfde bestand kunnen werken. Maar het probleem is niet dat men niet aan het zelfde bestand kan werken, maar dat er meerdere versies van het functioneel ontwerp ontstaan. Het gevolg van het geheel is dat erop meerdere plaatsen de wijziging moet worden ingevoerd.

De oorzaak en gevolgenanalyse zijn te lezen in bijlage B nader onderzoeksrapport.

5.3 Nader onderzoek

Het nader onderzoek is een vervolg van het vooronderzoek en maakt tevens gebruik van de gegevens die bekend zijn uit de interviews. In het nader onderzoek is de gewenste situatie beschreven en zijn er oplossingsideeën bedacht voor ieder knelpunt. De bedachte oplossingsideeën zijn in een kritische analyse matrix geplaatst om zo sterke verbanden te kunnen leggen, met de beschreven oplossing en daarbij het oplossingseffect op de knelpunten. Het nader onderzoek is afgesloten met een conclusie waarna aanbevelingen worden gedaan.

5.3.1 Gewenste situatie

Om de gewenste situatie te beschrijven is er gebruik gemaakt van de informatie uit de interviews. Het beeld dat ontstaan is uit de interviews is dat 'men het niet beter weet' men weet niet de oplossing. De gewenste oplossing zou zijn dat de problemen die er zijn en voornaamste het probleem van het beheer opgelost wordt. Om de gewenste situatie te beschrijven is er een onderscheid gemaakt tussen de wensen van de architecten en die van de software/test engineers omdat ze van elkaar verschillen. Ook is het zo, dat er toch andere belangen zijn, de architect is degene die het functionele ontwerp opstelt en de engineers maken er verder gebruik van.

Architecten

De architecten wil een tool die het mogelijk maakt om afbeeldingen (mockups) te maken zoals het nu gebeurt in MS Visio. Alleen hierbij wel de mogelijkheid om wanneer men één onderdeel in de afbeelding bewerkt, zeg als voorbeeld de header, dat die wijziging ook wordt gedaan in alle afbeeldingen die dit onderdeel heeft, in dit voorbeeld dan de header.

Ook is er een probleem met de verschillende versies van het functioneel ontwerp, het meest wenselijke zou zijn als er bijvoorbeeld bepaald kan worden voor welke versie de zojuist uitgevoerde wijziging is. Als de architect inmiddels in versie 3.6 werkt en er komt een wijziging voor 3.5, dan zou het wenselijke zijn als men die wijziging kan doorvoeren in beide versies. Dit door op een of andere manier in de versie waar de architecten nu mee bezig zijn te kunnen aangeven dat deze wijziging ook doorgevoerd moet worden in de voorgaande versies. Dit heeft als reden dat versie 3.5 wordt gekopieerd door de architecten om versie 3.6 in dit geval te creëren. Op het moment dat er wijzigingen zijn voor 3.5 dan zal die wijzigingen in beide versies doorgevoerd moeten worden, dit willen ze automatiseren.

Software Engineers / Test Specialist

Wanneer het functioneel ontwerp gewijzigd wordt na de Design meeting willen de software engineers en test specialisten duidelijkheid over wat er nu precies is gewijzigd in de vernieuwde versie. Dit zou bijvoorbeeld kunnen met een revisie tabel in het functioneel ontwerp onder te brengen, die aangeeft welk pagina's/hoofdstukken zijn aangepast.

Het zou ook wenselijk zijn als het duidelijk is wanneer er wijzigingen worden doorgevoerd in het functioneel ontwerp. Dit kan bijvoorbeeld door wijzigingen in het functioneel ontwerp op vaste tijden uit te voeren of een planning op te stellen. Dit reflecteert naar een knelpunt die niet zo'n hoge prioriteit heeft in dit project.

5.3.2 Oplossingen

Om de oplossingen in deze fase van het project al te benoemen is nog wel heel vroeg. Wel is er al nagedacht over de oplossingsideeën (gebrainstormd). Dit is afkomstig van de V2 methode en geeft al een redelijk inzicht in de komende oplossingsrichtingen. Het is zo opgezet dat voor ieder knelpunt oplossingen worden bedacht, zo kan het zijn dat een knelpunt 1 of meerdere oplossingsideeën heeft, in het geval van dit project is het laagste 2 oplossingsideeën. Op de volgende pagina is een Tabel 2 - Kritische analyse toegevoegd. De kritische analyse is ontleend aan de Quality Function Deployment methodiek waarbij de problemen/knelpunten uit de huidige situatie aan oplossingsideeën uit de gewenste situatie door middel van een matrix worden vergeleken. De reden voor het opstellen van zo'n matrix is dat meerdere oplossingsideeën positieve of negatieve invloed kunnen hebben op de gevonden problemen/knelpunten. Het kan dus zijn dat één oplossingsidee meerdere problemen/knelpunten oplost. De uitgeschreven oplossingsideeën zijn te vinden in de bijlage B nader onderzoeksrapport.

Tabel 2 - Kritische analyse

↓ Oplossingen	Problemen →	P1 Verschillende versies van FO	P2 Wijzigingen FO niet direct zichtbaar	P3 FO bestand is groot	P4 Aanpassen FO kost veel tijd/werk	P5 FO niet compleet
01.	Bewerken van het FO vereenvoudigen	x			x	
02.	Duplicatie van het FO zo min mogelijk houden	x			x	
03.	Samenwerken aan FO	x				
04.	Wijzigingen bijhouden		x/o			
05.	Beter bestand opslag met versiebeheer	x		x	x	
06.	Afbeeldingen (mockups) vereenvoudigen			x	x	o
07.	FO vaste tijden bewerken/publiceren.					x
08.	Reviewen van FO					x
09.	Bestand splitsen in master- en sub document.	x	x	o		
010.	SE en TE eerder betrekken bij de ontwikkeling				x	x
011.	Checklist opstellen om fouten te voorkomen					x/o
012.	Versturen van notificaties		x			
Prioriteit (1 hoog / 4 laag)		1	2	3	1	3

Legenda: x geeft aan dat de oplossing het probleem op zal kunnen lossen
 o geeft aan dat het probleem misschien zelfs verergert

5.3.3 Conclusie

In het voor- en nader onderzoek is de huidige situatie (ist) en gewenste situatie (soll) behandeld met behulp van de V2 methode. De conclusie van het onderzoek beschrijft dat er meer onderzoek nodig is om het probleem op te lossen en zeker ook om te voldoen aan de vragen die zijn gesteld in de PID.

Ook was duidelijk dat om de knelpunten op te lossen en de oplossingsideeën uit te voeren, dat daar software voor nodig is. Dit was al gestuurd door de opdrachtgever van het project door het stellen van de vraag of er software is voor het fabriceren/beheren van een ontwerpdocument, maar uit de resultaten van het initieel onderzoek kan ook degelijk geconcludeerd worden dat software een degelijk oplossingsrichting is.

De oplossingsrichtingen komen uit op de volgende categorieën (onderzoeksrichtingen): Deze categorieën zijn gevonden door onderzoek te doen en te kijken wat erin de markt beschikbaar is op het gebied van beheren/fabriceren van een functioneel ontwerp en die overeenkomen met de al bedachten oplossingsideeën.

Tools voor het fabriceren van een software product:

- **CASE-tools (Computer-aided software engineering)**
Deze tools kunnen een systeem vaak in technische detail beschrijven.
- **Requirement Management tools**
Deze tools geven de mogelijkheid om geordend requirements te beheer maar bieden vaak ook de mogelijkheid om requirement traceerbaar te maken.
- **Rapid Prototyping tools (Mockups/Wireframes) of Graphical user interface builder**
Deze tools maken het mogelijk Mockups en Wireframes te maken net als MS Visio alleen dan vaak de mogelijkheid om een werkend prototype te genereren.

Systemen voor het opslaan van het functioneel ontwerp:

- **ECM/DMS systeem (Enterprise Content Management, Document Management System)**
Dit systeem kan er voorzorgen dat er juiste versiebeheer/bestandsbeheer van documenten plaatsvindt.
- **Wiki**
Een Wiki is een database van artikelen die beheerd kunnen worden met uitgebreid versiebeheer.

De gestelde categorieën staan niet vast maar bieden een leidraad voor verder onderzoek. Mocht er tijdens het verdere onderzoek gestuit worden op een ander product kan er uiteraard ook besloten worden die mee te nemen in het onderzoek.

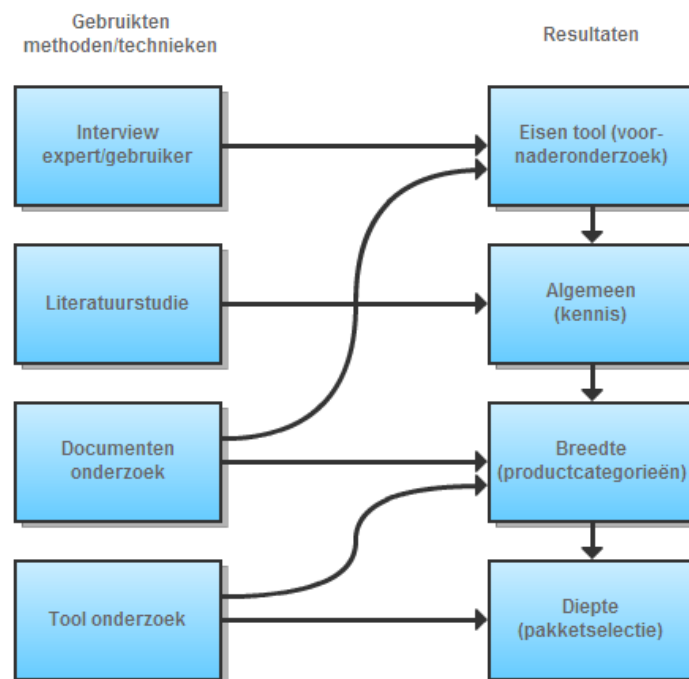
6 Diepteonderzoek

6.1 Inleiding

In dit hoofdstuk wordt het diepteonderzoek besproken. Eerst wordt het onderzoeksmodel uitgelegd en vervolgens komen de hoofdstukken terug zoals ze zijn in het diepteonderzoek.

6.2 Het Onderzoek

Het diepteonderzoek kent een stuk literatuur onderzoek en een stuk toolonderzoek waar de pakketselectie een onderdeel van is. Het onderzoeksmodel is gebaseerd op de gestelde vraagstellingen die in het begin zijn vastgesteld in de PID (Project Initiation Document). Zie Figuur 6 - Onderzoeksmodel.



Figuur 6 - Onderzoeksmodel

▪ Methoden van informatieverzameling

Interview expert/gebruiker

Tijdens het initieel onderzoek zijn er interviews gehouden met de betrokkenen om zo een juist beeld te krijgen van de huidige en gewenste situatie binnen Infoland.

Literatuurstudie

Om een theoretische basis te vormen is er een literatuurstudie uitgevoerd, ook wel literatuuronderzoek genoemd. Dit onderzoek bestaat uit informatie verzamelen over bepaalde ontwerpen van het internet, boeken, tijdschriften, artikelen en nog meer wat gevonden kan worden over het onderwerp. De gevonden informatie zal bewaard worden zodat er later op terug gevallen kan worden.

Documentenonderzoek

Om een beter beeld te krijgen van de huidige situatie is er ook gekeken naar de interne documenten van Infoland. Documenten als het functioneel ontwerp maar ook interne registraties als ToDo lijsten en Unit lijsten. Dit onderzoek is gedaan in het begin om op de juiste manier de interviewvragen op te stellen, maar ook om tijdens het schrijven van het onderzoek de huidige situatie te controleren.

Tool onderzoek

De tool onderzoek is een pakketselectie en kennisverzameling. Een zoektocht naar tools via het internet en denkbare informatie over de tools. Hierbij kan het gaan om kennis van de fabrikant zijn website maar ook kennis die eromheen beschikbaar is. Zo ook zoeken naar referenties en kennis hoe andere bijvoorbeeld met de tool omgaan.

- **De onderzoeksresultaten**

Eisen tool

Door kennis uit het voor- en nader onderzoek waar de knelpunten duidelijk zijn geworden kunnen systeemeisen worden ontleed. Deze systeemeisen kunnen gebruikt worden voor de pakketselectie om zo 't juiste softwarepakket te selecteren op basis van deze criteria. Ter verificatie en aanvulling is tijdens het onderzoek de systeemeisen doorgestuurd naar de architecten.

Algemeen

Door middel van de literatuurstudie is er een theoretisch kader opgesteld. De kennis dat opgedaan is uit de literatuurstudie is beschreven in de algemene hoofdstukken van het onderzoeksrapport.

Breedte

Met de kennis van het theoretische kader en het documentenonderzoek is er een beeld van de mogelijkheden en een gevoel voor de huidige situatie. Om te voorkomen dat er te snel de diepte in gegaan wordt, wordt er eerst in de breedte gekeken naar mogelijke tools die het probleem kunnen oplossen als in zijnde productcategorieën. Door kennis uit het toolonderzoek te combineren kan er een keuze gemaakt worden welke productcategorie diepgaand onderzocht moet worden.

Diepte

Er is een keuze gemaakt welke productcategorie diepgaand onderzocht gaat worden. In deze productcategorie wordt er specifiek gekeken naar iedere tool en in combinatie met de tool eisen wordt er een pakketselectie uitgevoerd. Deze pakketselectie heeft geresulteerd in 3 softwarepakketten die nader onderzocht zijn.

6.3 Functioneel Ontwerp

Uit het literatuuronderzoek is gebleken wat er in het functioneel ontwerp wordt ondergebracht. Het is van belang om te weten wat een functioneel ontwerp nu eigenlijk is en waarvoor het bedoeld is, omdat zodoende er een beter antwoord gegeven kan worden op de vraag hoe men zo'n functioneel ontwerp op de juiste manier kan beheren.

Uit dit onderzoek kunnen de volgende vragen beantwoord worden:

- Waar ontstaat een functioneel ontwerp?
 - Waarom het schrijven van een functioneel ontwerp?
 - Wie schrijft het functioneel ontwerp?
- Inhoud van een functioneel ontwerp?
- Hoe het functioneel ontwerp beheren?

Dit onderzoek is uitgevoerd via het internet met zoektermen als 'functioneel ontwerp', 'ontwerpdocument', 'functional design', 'design document', 'softwareontwikkelmethode', 'UML' en nog veel meer zoektermen. De resultaten, die de zoektermen hebben opgeleverd zijn artikelen, boeken, tijdschriften en voorbeelden van functioneel ontwerpen. De resultaten van het onderzoek zullen hieronder summier besproken worden per vraag.

6.3.1 Introductie

Wanneer men gaat beginnen met het ontwikkelen van software is het van belang dat er van te voren wordt nagedacht van wat er ontwikkeld gaat worden. Wanneer men dit niet doet zal er veel tijd verloren gaan tijdens de daadwerkelijke ontwikkeling.

Daarom kennen alle softwareontwikkelmethodes de volgende opzet:

1. Requirement Analyse / Bepalen van de eisen en wensen
2. Ontwerpen van het systeem
3. Ontwikkelen van de software (programmeren)
4. Testen van het systeem

In de eerste stap zal men gaan bepalen waaraan het systeem moet voldoen. Dit kan op verschillende manieren gedaan worden. Deze stap zal voornaamste invloed hebben op het ontwerpen van het systeem. In stap twee wordt het systeem ontworpen hier wordt vaak het functioneel ontwerp opgesteld. In het functioneel ontwerp staat wat de te bouwen applicatie of website aan functionaliteit moet bieden zodanig omschreven dat vaak een klant het document ook goed kan begrijpen. Stap drie wordt het systeem ontwikkeld door de ontwikkelaars en in stap vier wordt het systeem vervolgens getest.

Waarom het schrijven van een functioneel ontwerp?

Een belangrijk voordeel van het schrijven van een functioneel ontwerp is het stroomlijnen van het ontwikkelingsproces. De ontwikkelaars en testers kunnen het document gebruiken om het systeem te maken, testen of een inschatting te maken qua planning. Wat ook belangrijke bijdragen kan zijn, is dat het functioneel ontwerp als communicatiemiddel gebruikt kan worden met de opdrachtgever, om zo in overleg te kunnen bepalen hoe het systeem eruit komt te zien.

Wie schrijft het functioneel ontwerp?

Het functioneel ontwerp kan het beste geschreven worden door een specialist die niet betrokken is bij andere aspecten van het project¹. Die persoon moet aandacht hebben voor detail en zich goed kunnen inleven in de toekomstige gebruikers. Ook moet men voldoende kennis van technologie hebben om de beperkingen en mogelijkheden in te kunnen schatten. Men zal verschillende rollen moeten kunnen vervullen zoals klant, opdrachtgever, ontwikkelaar enzovoort en ideeën kunnen presenteren.

6.3.2 Inhoud functioneel ontwerp

De inhoud van een functioneel ontwerp kan verschillen. Dit kan afhangen van wat voor systeem er gemaakt wordt, de doelgroep en hoe andere mensen er mee omgaan. De diepgang van het functioneel ontwerp kan ook afwijken van de omvang van het systeem.

De meeste voorkomende onderdelen in het functioneel ontwerp zijn:

- Introductie van wat het systeem moet doen;
- Lijst met eisen en wensen/requirements die van toepassing zijn;
- Beschreven in- en uitvoer van het systeem;
- User interface (gebruikersinterface) met toelichting;
- Flowchart m.b.t. het gedrag van het systeem, storyboards, sitemap, state transition diagrams et cetera.

Ontwikkelmethodes en functioneel ontwerp

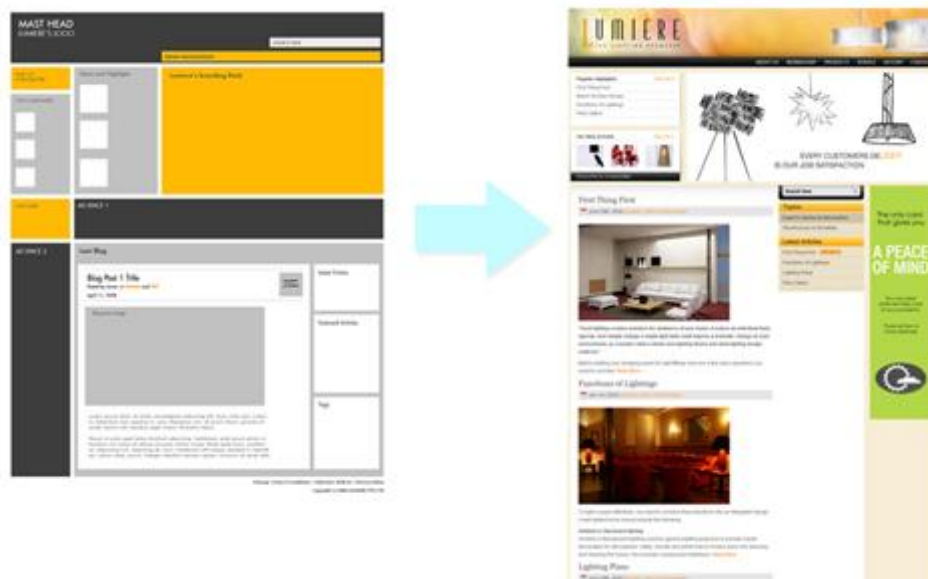
Het functioneel ontwerp (FO) kan qua inhoud behoorlijk verschillen en dit kan komen door de gebruikte ontwikkelmethode. Zo wordt erin het watervalmethode een heel gedetailleerde FO opgesteld, in het begin, voordat er geprogrammeerd wordt. In een methode als SCRUM wordt er bijvoorbeeld gewerkt met alleen user stories.

¹ Smith, A.W., 2001, mojafat.com



Wireframes vs. Mockups

Het functioneel ontwerp bevat meestal altijd een illustratie hoe de gebruikersinterface eruit komt te zien. Deze gebruikersinterface kan op verschillende manieren opgesteld worden maar komen altijd neer op een mockup of een wireframe. Een mockup is een schermafbeelding van de nog te realiseren systeem, maar wel een afbeelding dicht tegen de werkelijkheid aan. Daar tegen over staat een wireframe dat hetzelfde is als een mockup alleen dan met minder detail, een wireframe kan bijvoorbeeld al een schets op papier zijn van de nog te ontwikkelen systeem. In Figuur 7 is het verschil te zien tussen wireframes en mockups, links het wireframe en rechts de mockup.



Figuur 7 - Wireframe vs. Mockup

Beiden hebben ze een doel ten aanzien van communicatie. Ten aanzien van een mockup zal bijvoorbeeld de klant meer letten op gebruikte lettertype/kleuren enzovoort waarbij het wireframe meer nadruk gelegd wordt op de locatie van elementen.

Standaarden

Voor onderdelen in het functioneel ontwerp zijn er standaarden opgesteld als bijvoorbeeld UML (Unified Modeling Language) en BPMN (Business Process Modeling Notation). Het voldoen aan standaarden kan voordeel opleveren en softwarepakketten adopteren vaak zulke standaarden. De inhoud van een functioneel ontwerp zal daar vervolgens op afgestemd moeten worden.

6.3.3 Beheer functioneel ontwerp

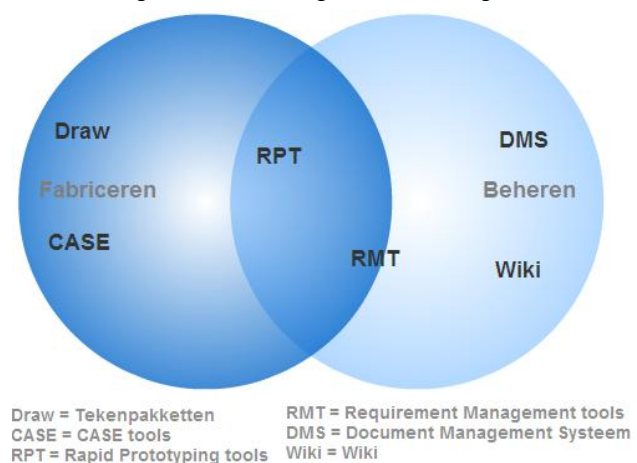
Het beheer is heel erg afhankelijk van wat hiervoor staat beschreven. Maar het beheren van het functioneel ontwerp is heel belangrijk en zeker bij de watervalmethode die Infoland gebruikt. In een ideale situatie zou het functioneel ontwerp bevroren moeten worden wanneer die wordt overgedragen aan de ontwikkelaars. Alleen het blijft onoverkoombaar dat er naar de overdracht wijzigingen ontstaan. Daarom moet het functioneel ontwerp goed nagekeken worden en regelmatig van updates worden voorzien.

6.4 Productcategorieën

In de conclusie van het nader onderzoek kwam na voren dat er een oplossing in de richting van een softwareproduct gezocht moest worden. Ook dit komt overeen met de vraag of er software is voor het beheren/fabriceren van een functioneel ontwerp. In het nader onderzoek waren al productcategorieën genoemd die mogelijk een oplossing bieden. Om niet direct de diepte in te gaan is er besloten om eerst in de breedte te kijken wat de mogelijkheden zijn. Dit heeft een zoektocht opgeleverd naar producten in de verschillende productcategorieën en hier van is een lijst van bijgehouden in MS Excel. Zie hiervoor bijlage C Software Lijst.

De productcategorieën die het beheer/fabriceren van het functioneel ontwerp kunnen vereenvoudigen zijn elk beschreven met de volgende informatie: welke mogelijkheden biedt zo'n product, waar is het product voor gediend, wie maakt er over het algemeen gebruik van en wat zijn de voordelen van zo'n product. Vanuit de gestelde vraag en de visie naar de oplossing zou het pakket in beide verzamelingen (fabriceren en beheren) thuis moeten horen. Om dit te kunnen visualiseren is er gebruik gemaakt van een venndiagram waarbij de twee verzamelingen getekend zijn als cirkels, en de productcategorieën zijn daarop geplaatst zie Figuur 8.

Figuur 8 - Productcategorieën in venndiagram

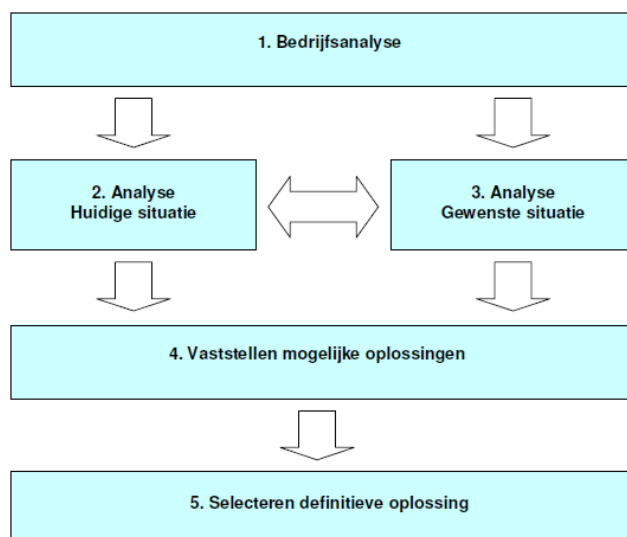


De productcategorieën die links en rechts staan kunnen gezien worden als ondersteunende producten. Dit omdat ze de knelpunten binnen het project niet direct zullen oplossen maar wel degelijk indirect kunnen bijdrage aan het succes. Binnen het project zijn de producten in het midden $A \cap B$ datgene waarna gekeken kan worden. Dit zijn RPT (Rapid Prototyping tools) en RMT (Requirement Management tools) waarbij RMT nog op de rand begeeft.

Om nog beter onderscheid te maken tussen de productcategorieën zijn de producten nog vergeleken met de knelpunten. Dit is in beeld gebracht door het gebruik van een matrix. In de matrix is aangegeven of een product het knelpunt zal verhelpen of kan verminderen. Hieruit bleek zoals in het venndiagram ook al duidelijk werd dat RPT (Rapid Prototyping tools) en RMT (Requirement Management tools) heel dicht bij elkaar in de buurt komen. De keuze is gevallen om verder onderzoek te doen naar RPT pakketten, dit omdat deze productcategorie de meeste knelpunten lijkt te gaan oplossen en het komt 't dichtste bij de huidige de geschetste gewenste situatie in de buurt. RMT producten bieden ook mogelijkheden in het project alleen staan nog het verste af van de huidige situatie bij Infoland. Dit komt doordat het functioneel ontwerp het meest leidende stuk is in het proces en requirements eigenlijk niet veranderen.

6.5 Pakketselectie

Om tot het beste softwarepakket te komen voor Infoland die hopelijk de knelpunten zal wegnemen is de keuze gevallen om een pakketselectie uit te voeren. Als men zoekt naar pakketselectie methode op internet komt men uit op methode als: PSI (pakketselectie en -implementatie methode), Indora en KPMG. PSI is een heel uitgebreide methode voornamelijk als er maatwerk bij betrokken is en koppelingen met andere systemen.



Figuur 9 - Indora pakketselectie methode

Daarom valt PSI hierbij al af omdat dit niet het uitgangspunt is van dit onderzoek. Indora weergegeven in Figuur 9 en KPMG zijn heel vergelijkbare methodes beide werken ze met de stappen (termen) als: vooronderzoek, longlist en shortlist. De keuze is gevallen om Indora te gebruiken als hulpmiddel. Indora heeft duidelijke gelijkenissen met de V2-methode die tevens wordt gebruikt als hulpmiddel. De duidelijke scheiding tussen het analyseren van de huidige situatie en gewenste situatie is bijvoorbeeld gebeurd in het voor- en nader onderzoek. KPMG daarentegen legt minder nadruk op het vooronderzoek dan Indora. Ook al ziet KPMG het vooronderzoek als een kritieke stap bevat het niet de diepgang zoals Indora illustreert.

6.5.1 Methode

De methode die gekozen is ter ondersteuning voor de pakketselectie is Indora. De methode kent de volgende stappen:

1. Bedrijfsanalyse
2. Analyse Huidige situatie
3. Analyse Gewenste situatie
4. Vaststellen mogelijke oplossingen
5. Selecteren definitieve oplossing
 - a. Longlist en knock out criteria
 - b. Shortlist en detailonderzoek
 - c. Offerte-aanvraag en keuze

De eerste 3 stappen zijn al uitgevoerd op basis van V2 methode en TSP en is te vinden in hoofdstuk 5 "Initieel onderzoek". Stap 4 kent het spreekwoordelijke gezegden "*Vele wegen leiden naar Rome*". Deze stap is uitgevoerd in dit diepteonderzoek waarbij het duidelijk is dat er gericht gekeken moet worden naar softwarepakketten. Dit is zodanig uitgevoerd dat erin de breedte is gekeken naar mogelijke productcategorieën hoofdstuk 6.4. Tevens is erin de markt gekeken naar mogelijke software oplossingen, en daaruit is een analyse uitgevoerd met basiscriteria waaruit een matrix is opgesteld. Stap 5 is het daadwerkelijk uitvoeren van de pakketselectie. Hier is de keuze gemaakt om het als volgt uit te voeren: stap a, door middel van knock out criteria softwarepakketten uit te kiezen waarvoor een demo wordt aangevraagd, stap b, detailonderzoek van het handjevol overgebleven softwarepakketten.

6.5.2 Analyse

Deze analyse is uitgevoerd om op een lijst te komen van Rapid Prototyping tools (Mockups/Wireframes) en Graphical user interface builder tools. De zoektocht naar softwarepakketten is gedaan via internet om zo tot een lijst van pakketten te komen, deze lijst is bijgehouden in MS Excel. De zoektermen die gebruikt zijn om tot de softwarepakketten te komen zijn bijvoorbeeld 'Mockup tools' en 'Rapid Prototyping Tools'. Die zoektermen zullen resulteren in lange lijsten van beschikbare tools. Ook zijn zoektermen gebruikt in combinatie met gevonden producten stel het pakket Axure en iRise zijn zojuist gevonden daaruit kunnen bijvoorbeeld zoektermen opgesteld worden als 'Axure Review' en 'Axure vs iRise'. Die zoektermen leveren vaak artikelen op van deskundige en tevens



reacties van andere deskundige, en soms ook weer linkt naar andere artikelen. Vervolgens is die lijst geanalyseerd waarbij softwarepakketten die alleen beschikbaar zijn voor een Mac uit de lijst zijn gehaald. Ook zijn softwarepakketten waar niet meer actief aan wordt door ontwikkeld geschrapt van de lijst. Vervolgens blijven er 25 softwarepakketten over die onder de loep worden genomen door middel van 11 basiscriteria.

Basiscriteria

Deze basiscriteria zijn vrij uniek omdat het hierbij gaat om een bepaalde productcategorie. Deze criteria zijn gedeeltelijk afkomstig van vergelijkingssites die artikelen hebben geschreven over deze productcategorie. Andere criteria zijn sterk ontstaan vanuit de behoefte gericht op de gewenste situatie. Hieronder de opsomming van de 11 criteria:

- Platform (waarop het softwarepakket beschikbaar is zoals Windows of Online)
- Drawing diagrams (geeft aan of het systeem de mogelijkheid biedt om uitgebreid diagrammen te teken zoals het maken van een dergelijke flowchart)
- Wireframe / Mockups (mogelijkheid om wireframes of mockups te tekenen)
- Prototype (biedt het softwarepakket de mogelijkheid om de wireframes/mockups om te zetten naar een prototype)
- Master layers / pages (de mogelijkheid om veel terugkomende onderdelen zoals headers, navigatie en footers te kunnen hergebruiken)
- Export format (welke specifieke export formaten zijn er)
- Output documentatie (mogelijk is om een Word document te genereren)
- Collaboration (mogelijkheid tot het samenwerken)
- Change Management (worden wijzigingen weergegeven en kunnen ze worden teruggedraaid)
- Reviewing (mogelijkheid om commentaar te ontvangen)
- Prijs (de prijzen van alle pakketten)

In Tabel 3 die zich bevindt op de volgende pagina is de matrix opgenomen. Hier worden de 25 gevonden softwarepakketten geanalyseerd aan de hand van de zojuist beschreven basiscriteria. De informatie die in de matrix is verwerkt is afkomstig van de fabrikant zijn website, de vergelijkingssites en de gevonden artikelen/reviews. Meeste fabrikanten boden demo filmpjes of uitgebreide tekstuele informatie aan op de website. Het was niet nodig om te werk te gaan met een Request for Information (RFI) formulier zoals beschreven wordt in Indora's pakketselectie. Dit is een vragenlijst dat naar de fabrikant wordt gestuurd om antwoord te krijgen of het product voldoet aan de gestelde criteria. Doordat in deze productcategorie de informatie goed toegankelijk wordt gemaakt op de website van de fabrikant, is het niet nodig om de fabrikant te contacteren.

Matrix

Hieronder Tabel 3 de ingevulde matrix.

Mockup Tools												
	Platform	Drawing diagrams	Wireframe	Mockups	Prototype	Master layers \ Pages	Export format	Output documentatie	Collaboration	Change Management \ Track Changes	Reviewing	Notities
Gliffy	Online	v	v			HTML						Gratis Algemene tool voor diagrammen
Lumzy	Online	v	v*	v	v	PDF		v				Gratis Open wireframe tool
Pencil	Firefox	v	v			Many						Gratis Opensource firefox add-on
Pidoco	Online	v	v*	v	v	HTML	v	v	v*	v		€ 7 Prijs per maand
GoMockingbird	Online	v				PNG		v				€ 9 Prijs per maand
JustProto	Online	v	v	v	v	HTML		v				€ 14 Prijs per maand
iPlotz	Online	v		v	v	HTML						€ 15 Prijs per maand
Cacoo	Online	v	v*			Many		v		v		€ 25 Prijs per maand
Creately	Online	v	v*			Many		v				€ 25 Prijs per maand
HotGloo	Online	v	v	v		PNG		v				€ 28 Prijs per maand
Lovelycharts	Win, Mac	v	v			Many						€ 59 Algemene tool voor diagrammen
MockFlow	Win, Mac	v			v	Many	v	v				€ 59 Output documentatie kost extra
Balsamiq	Win, Mac	v	v			PNG						€ 79 Wireframe tool
DesignerVista	Win	v	v	v		Many	v					€ 80 Uitgebreide mockup tool
Flairbuilder	Win, Mac	v			v	PNG						€ 99 Wireframe tool
Wireframesketcher	Win, Mac	v	v			PDF		v				€ 99 Eclipse add-on
Mockup Screens	Win, Mac	v	v*			Many	v					€ 250 Goed voor kleine projecten
Lucid Spec	Win		v	v		PNG	v					€ 299 Oude look en feel
Justinmind	Win, Mac		v	v	v	Many	v					€ 495 Uitgebreide mockup tool met prototypes
Carretta Gui Design	Win		v	v	v	Many	v	v				€ 499 Uitgebreide mockup tool met prototypes
Visio	Win	v	v	v*		Many						€ 560 Algemene tool voor diagrammen
Axure	Win, Mac	v*	v	v	v	Many	v	v	v*	v*		€ 589 Uitgebreide mockup tool met prototypes
SketchFlow	Win, Mac		v	v	v	HTML	v					€ 599 Add-on voor Microsoft Expression
ProtoShare	Online	v	v	v	v	HTML	v	v				€ 950 Uitgebreide mockup tool met prototypes
iRise	Win, Mac	v*		v	v	Many	v	v	v	v		€ 6.995 Enterprise prototype tool

* = Voldoet gedeeltelijk

Tabel 3 - Software Analyse Matrix

6.5.3 Longlist

De matrix vergelijkt 25 softwarepakketten en uit die pakketten zijn er 12 gekozen waarvoor een demo is aangevraagd. De keuze voor een demo in plaats van criteria is om een nog beter beeld te krijgen van wat de softwarepakketten kunnen. In de demo wordt het pakket uitgetest en worden basishandelingen uitgevoerd om een vergelijkbaar scherm te creëren als in de functionele ontwerpen van Infoland. Wel is hier rekening gehouden met vertrouwelijke materialen omdat de ontwerpen in context vertrouwelijk zijn. Tijdens de demo worden de pluspunten en de minpunten bijgehouden (pro's en con's) net als bij "Decision making" methodes.

De volgende softwarepakketten zijn gekozen voor de demo's: Pencil, Pidoco, JustProto, Hotgloo, DesignerVista, Lucid Spec, Justinmind, Gui Design, Axure, Sketchflow, ProtoShare en iRise. De keuze voor deze 12 softwarepakketten bestaat uit het feit dat 9 pakketten volledige mockup ondersteuning bieden. Van de andere 3 pakketten is er één open source en de andere voldoen gedeeltelijk (meer dan de overblijvers) aan de gestelde basiscriteria. De keuze voor het open source pakket is om de variantie te ervaren omdat het hierbij gaat om een Firefox add-on.

Pencil (versie 1.3-4)	Pidoco (versie maart 2012)
<ul style="list-style-type: none"> + Eenvoudig in gebruiken + Open source - Het is een Firefox add-on - Export is minimaal - Geen masters 	<ul style="list-style-type: none"> + Online + Overzichtelijk projectoverzicht - Geen gedetailleerde mockups - Werkt met statische layers. Layers zitten vast op één positie. Vergelijkbaar met Photoshop
JustProto (versie maart 2012)	Hotgloo (versie 3.0)
<ul style="list-style-type: none"> + Eenvoudig in gebruik + Online - Geen mogelijkheid om eigen elementen (widgets) toe te voegen. - Geen documentatie alleen HTML prototype - Maar één master per pagina 	<ul style="list-style-type: none"> + Uitgebreid versiebeheer + Online - Beperkt aantal elementen - Geen documenten output
DesignerVista (versie 5.0)	Lucid Spec (versie 1.6)
<ul style="list-style-type: none"> + Uitgebreide basiselementen + Document generator goed + Office look en feel - Geen prototype - Geen masters - Bij 't opstarten elke keer vragen wat voor applicatie er gemaakt moet worden. (web, ribbons etc.) 	<ul style="list-style-type: none"> + Helpdesk reageert snel - Werkt niet krijg foutmeldingen

Justinmind (versie 4.5) <ul style="list-style-type: none"> + Uitgebreid + Kan werken met een datagrid - Moeilijk diagrammen tekenen - Technische aspecten waarvoor training nodig is 	Caretta Gui Design (versie 4.2.116.0) <ul style="list-style-type: none"> + Kan updates van documentatie creëren - Opties zitten onlogische plaatsen - Masters werken niet naar behoren - Totaal ervaring niet echt fijn
Axure (versie 6.0.0.2913) <ul style="list-style-type: none"> + Uitgebreid + Aanpasbare en goede documentatie output + Hulpvaardige Community + API voor output - Technische aspecten waarvoor training nodig is - Prototype genereren duurt langer, hoe groter het project wordt. - Werken met dynamische panelen voor gewenst resultaat. 	SketchFlow (versie 4) <ul style="list-style-type: none"> + Uitgebreid + Koppeling met andere Microsoft producten - Ingewikkeld - Lastig elementen plaatsen - Minder opties voor output documentatie
ProtoShare (versie 6.1) <ul style="list-style-type: none"> + Simpel + Online - Niet veel mogelijkheden voor kleuren te bepalen - Documentatie beperkt 	iRise (versie 8.9) <ul style="list-style-type: none"> + Documentatie + Koppeling met Requirement management tools mogelijk. + Uitgebreide prototyping - Geen juiste mogelijkheid om wireframe of mockups te creëren. - Er moet met afbeeldingen gewerkt worden. - Prijzig

6.5.4 Shortlist

Om nu tot een gedegen softwarepakket te komen is er gekeken naar de 12 pakket demo's in de longlist om zo vervolgens een shortlist op te stellen voor het detailonderzoek. De keuze is gevallen op 3 softwarepakketten namelijk Axure, Justinmind Prototyper en ProtoShare. Dit is gebaseerd op de plus- en minpunten die zijn gegeven aan de longlist. Het was niet nodig om gewichten te hangen aan de plus- en minpunten om tot een besluit te komen. Zo vallen softwarepakketten af die geen master layers kunnen creëren zoals gewenst, pakketten die te ingewikkeld waren en sommige boden niet de elementen om de mockups te creëren zoals gewenst.

Om het detailonderzoek uit te voeren zijn er 13 functionele systeemeisen opgesteld waar de softwarepakketten aan worden onderworpen. Elke eis zal toegelicht worden en krijgt een score van 1 tot en met 10 waarbij 10 het hoogste is. De toelichting zal niet verwerkt worden in dit verslag wel komt het resultaat terug in het volgende hoofdstuk.

Functionele systeemeisen

De softwarepakketten zijn getest met de volgende functionele systeemeisen; deze eisen zijn opgesteld voor de gewenste situatie door de architecten en de manager:

Vergelijkbaar met Visio

- Het softwarepakket moet vergelijkbare mogelijkheden hebben als MS Visio. Het gene moet kunnen 'drag-and-droppen' van elementen en bijvoorbeeld het kunnen tekenen van diagrammen of figureren.
- Tekeningen in één handeling kunnen opnemen in het ontwerpdocument.

Documentatie

- Het moet mogelijk zijn om een document te kunnen genereren met schermafbeeldingen en tekstuele beschrijving. Dit zo vergelijkbaar mogelijk met het bestaande functioneel ontwerp.

Versiebeheer

- Het moet mogelijk zijn om met versiebeheer te kunnen werken. Dit kan zijn op niveau van elementen, schermen of bestand. Dit zal dan ook de mogelijkheid bieden om versies terug te kunnen draaien mocht er iets fout zijn gegaan.

Samenwerken

- Er zal een mogelijkheid moeten zijn om samen te kunnen werken al dan niet real-time. Men zal dus met meerdere personen aan het ontwerp moeten kunnen werken.

Wijzigingen bijhouden

- Het softwarepakket zal de mogelijkheid moeten bieden om de wijzigingen bij te houden. Dit zou in verband gebracht kunnen worden met versiebeheer. Dit zou betekenen dat er achteraf nog mogelijk is te zien wat er allemaal gewijzigd is.

Gebruikersvriendelijk

- Het softwarepakket zal gebruiksvriendelijk moeten zijn. Zeker omdat men waarschijnlijk veel gebruik van het systeem gaat maken. Met gebruiksvriendelijk wordt verstaan dat het pakket eenvoudig en efficiënt is voor dagelijks gebruik.

Import functionaliteit voor al bestaande functionele ontwerpen

- Het zou eenvoudig zijn als het softwarepakket een mogelijkheid bied om al bestaande functionele ontwerpen te importeren.

Kan vergelijkbare mockups creëren

- Met het softwarepakket moet het mogelijk zijn vergelijkbare of betere mockups te creëren dan in de huidige situatie met MS Visio zijn gemaakt.

Beschikt over en API

- Infoland is een softwarebedrijf en het zal daarom ook wenselijk zijn als het softwarepakket de mogelijkheid bied om met een API aan de slag te gaan. Een API geeft de mogelijkheid om zelf functionaliteiten te creëren.

Mogelijk om twee versies in één te beheren

- Het zou eenvoudig zijn als het softwarepakket de mogelijkheid bied om twee versies van het zelfde functioneel ontwerp te beheren. Hierbij gaat het om de nog te onderhouden versie bijvoorbeeld 3.4 en de juiste versie die nu wordt gemaakt bijvoorbeeld 3.5.

Mogelijkheid tot ondersteuning

- Indien er problemen zijn moet er ondersteuning geleverd kunnen worden door de leverancier.

Mogelijkheid tot het creëren van masters layers/pages

- Het softwarepakket moet de mogelijkheid hebben om master layers/pages te maken.

Aanpasbare velden voor meer informatie

- Het softwarepakket moet de mogelijkheid bieden voor de gebruiker om zelf velden te kunnen toevoegen t.b.v. meer informatie. Hierbij kan bijvoorbeeld gedacht worden aan een koppeling nummer met requirements.

6.5.5 Resultaat

De 3 softwarepakketten zijn beoordeeld aan de hand van een score van 1 tot en met 10. Hier staat een 1 voor de laagst mogelijke score en een 10 als de hoogst mogelijke score. Iedere beoordeling is ook verklaard in het diepteonderzoek. Alleen in dit verslag worden de verklaringen weggelaten. Hieronder is Tabel 4 weergegeven van de scores die zijn gegeven voor ieder pakket. Hierbij geeft een onvoldoende aan dat het systeem totaal niet voldoet aan de functionele eis (1) of onvoldoende beheerst (<6). Alles boven de 5 geeft aan dat het systeem er iets voor biedt en een 10 geeft aan dat het volledige voldoet aan de functionele eis.

	Functionele systeemeisen	Axure	JIM	ProtoShare
1	Vergelijkbaar met Visio	10	9	7
2	Documentatie	8	4	6
3	Versiebeheer	7	4	5
4	Samenwerken	7	5	8
5	Wijzigingen bijhouden	6	6	6
6	Gebruikersvriendelijk	7	7	7
7	Import functionaliteit voor al gemaakte functionele ontwerpen	6	6	1
8	Kan vergelijkbare mockups creëren	8	8	6
9	Beschikt over een API	8	1	1
10	Mogelijk om twee versies in één te beheren	7	1	7
11	Mogelijkheid tot ondersteuning	10	10	10
12	Mogelijkheid tot het creëren van masters	10	10	10
13	Aanpasbare velden voor meer informatie	10	5	5
	Score	8	5,8	6,1

Tabel 4 - Scores Tools

Schaal Architecten

De architecten hebben ook bij elke functionele eis een schaalscore gegeven waarbij ze aangeven wat zij het belangrijkste vinden, zie Tabel 5. Hierbij geldt ook de score 1 tot en met 10.

	Functionele systeemeisen	Schaal
1	Vergelijkbaar met Visio	10
2	Documentatie	9
3	Versiebeheer	8
4	Samenwerken	8
5	Wijzigingen bijhouden	3
6	Gebruikersvriendelijk	9
7	Import functionaliteit voor al bestaande functionele ontwerpen	6
8	Kan vergelijkbare mockups creëren	4
9	Beschikt over een API	1
10	Mogelijk om twee versies in één te beheren	4
11	Mogelijkheid tot ondersteuning	2
12	Mogelijkheid tot het creëren van masters	4
13	Aanpasbare velden voor meer informatie	2
	Totaal	70

Tabel 5 - Schaalscores

Totaalscore

Nu de scores van Tabel 4 - Scores Tools en Tabel 5 - Schaalscores bekend zijn kan de totaalscore berekend worden. Dit wordt gedaan door middel van de gewogen factor score methode. De scores van de softwarepakketten worden vermenigvuldigd met de schaalscore waaruit een score van 1 tot en met 100 komt. De scores worden bij elkaar opgeteld gedeeld door 70, het totaal van de schaalscores van de architecten levert een rapportcijfer op voor ieder pakket. Zie Tabel 6.

	Functionele systeemeisen	Axure	JIM	ProtoShare
1	Vergelijkbaar met Visio	100	90	70
2	Documentatie	72	36	54
3	Versiebeheer	56	32	40
4	Samenwerken	56	40	64
5	Wijzigingen bijhouden	18	18	18
6	Gebruikersvriendelijk	63	63	63
7	Import functionaliteit voor al gemaakte functionele ontwerpen	36	36	6
8	Kan vergelijkbare mockups creëren	32	32	24
9	Beschikt over een API	8	1	1
10	Mogelijk om twee versies in één te beheren	28	4	28
11	Mogelijkheid tot ondersteuning	20	20	20
12	Mogelijkheid tot het creëren van masters	40	40	40
13	Aanpasbare velden voor meer informatie	20	10	10
	Scores (gesommeerd totaal / 70)	7,8	6,0	6,3

Tabel 6 - Totaalscores

Justinmind Prototyper (JIM) en ProtoShare zijn vergelijkbare producten. Het grote verschil is dat ProtoShare een online product is en alleen gebruikt kan worden in de browser. JIM en Axure werken voornamelijk lokaal. Uit de totaalscores dat de softwarepakketten gekregen hebben steekt Axure er bovenuit met een 7,8. De twee overige pakketten kregen een 6,0 en 6,3. Axure is daarom het pakket dat het beste aangeraden kan worden voor Infoland. De uitgebreide mogelijkheid om automatisch een functioneel ontwerp te creëren dat vergelijkbaar is met het huidige functioneel ontwerp zal tijd kunnen besparen. Het systeem zal het voornaamste voordeel bieden op het moment dat men wijzigingen moet doorvoeren. Men kan in de tool een aanpassing doen aan bijvoorbeeld de schermen en zo weer een nieuw functioneel ontwerp generen. Dit zal de up-to-date zijn van het functioneel ontwerp en het werkelijke product kunnen bevorderen. Alleen dit pakket zal niet alle knelpunten wegnemen maar kan wel wat extra's bieden.

6.6 Conclusie

Het diepteonderzoek is het vervolg van het nader onderzoek om zo de vragen als gesteld in de PID zo goed mogelijk te kunnen beantwoorden. Als eerst is gekeken naar de opzet van een functioneel ontwerp. Het functioneel ontwerp kan per project verschillen en is erop gericht om duidelijk te maken wat er gemaakt moet worden. Dit is dan vaak gericht op een klant en de ontwikkelaars die het vervolgens gaan maken. Maar de opzet van het functioneel ontwerp is niet bepalend of het functioneel ontwerp goed of fout is. Alles wordt bepaald door de inhoud en de doelgroep waarvoor het functioneel ontwerp is geschreven.

Ook is ter controle gekeken in de breedte na verschillende productcategorieën die kunnen bijstaan bij het fabriceren/beheren van een functioneel ontwerp. Hier is gekeken naar 6 productcategorieën die vervolgens in een venndiagram zijn geplaatst om zo onderscheid te kunnen maken en de 'beste' oplossing te kunnen kiezen. De uitkomst is de productcategorie Rapid Prototyping tools.

In de gekozen productcategorie is er onderzoek gedaan naar de mogelijke softwarepakketten en daaruit zijn 25 pakketten gekomen. Hieruit komt één pakket die het dichtste bij Infoland ligt namelijk Axure. Dit pakket biedt de mogelijkheid om de mockups dynamisch op te bouwen waardoor het beheer op het functioneel ontwerp geoptimaliseerd wordt. Doordat het pakket ook de oplossing biedt om het functioneel ontwerp automatisch te genereren blijft het document dynamisch. Axure heeft een actieve community die kan helpen bij vragen en er worden uitgebreide instructie filmpjes geboden op de website.

Dat Axure het dichtste in de buurt komt bij de werkwijze van Infoland is een goed teken, wel moet nog onderzocht worden of Axure past bij Infoland in de huidige problematiek. Het is duidelijk dat Axure enkele knelpunten kan wegnemen maar tijdens het onderzoek is toch wel zichtbaar dat er wat haken en ogen aan de oplossing zitten. Het is te adviseren om verder onderzoek te doen naar de oplossingsrichtingen en daarbij dient gekeken te worden hoe Axure daar in past, dit wordt uitgevoerd in het oplossingsplan.

7 Oplossingsplan

7.1 Inleiding

Het diepteonderzoek heeft geresulteerd in een softwarepakket dat het beste past bij de eisen en wensen van Infoland. Maar toch bleek in de ingerichte demo en voordracht dat er nog wat haken en ogen aan de oplossing zitten. Het oplossingsplan is gebruikt om duidelijk te krijgen wat er precies mis is met Axure. Ook worden er verdere oplossingen aangedragen die kunnen bijstaan bij het oplossen van de knelpunten.

7.2 Knelpunten

Het oplossingsplan heeft één hoofdstuk genaamd knelpunten hier worden de knelpunten zoals ze zijn gedefinieerd in dit project beschreven. Ieder knelpunt bevat een omschrijving afkomstig van het voor- en nader onderzoek. In het oplossingsplan zijn de oorzaken die per knelpunt zijn gedefinieerd nadrukkelijk beschreven. De reden voor deze nadrukkelijke beschrijving is dat dit als opstap gebruikt kan worden voor eventueel vervolg projecten die bij Infoland plaats kunnen vinden. Ook zal het een duidelijk beeld geven van de situatie voor de lezer van het oplossingsplan. Ook worden de knelpunten en oorzaken verder in het document gebruikt. Zo kan bijvoorbeeld de knelpunten/oorzaken geraamd worden op Axure om te zien of Axure de oorzaken wegneemt. Voor de beschrijvingen van de knelpunten/oorzaken zie bijlage D knelpunten (oplossingsplan).



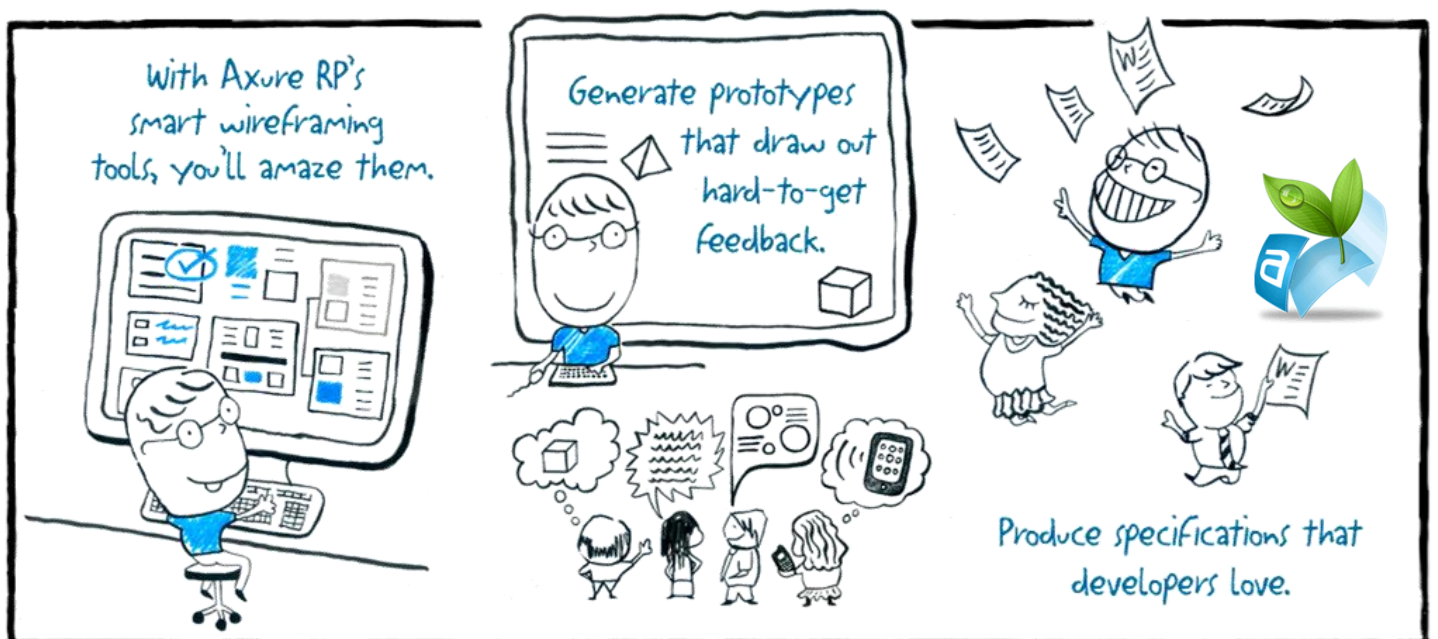
7.3 Software oplossing

Het mag duidelijk zijn dat Axure de software oplossing is dat uit de pakketselectie is gekomen. In dit hoofdstuk van het oplossingsplan is Axure geraamd op de situatie bij Infoland. Dit is gedaan door per knelpunt de voor- en nadelen te onderzoeken. Aan de hand van die informatie is er een matrix opgesteld.



Figuur 10 - Logo Axure

Axure biedt de mogelijkheid om vergelijkbare mockups te creëren als nu gebeurt in MS Visio. Buiten het creëren van mockups kan Axure ook veel meer. Men kan werken met master layers dat mogelijk maakt elementen die herhaaldelijk gebruikt worden op een centrale plaats te bewaren, wanneer het element (master layer) veranderd wordt, verandert het op iedere locatie waar dit element gebruikt wordt. Ook biedt Axure de mogelijkheid om een prototype van het ontworpen systeem te generen. Dit prototype is geheel interactief dus men kan bijvoorbeeld een formulier invullen en op links klikken. Maar het belangrijkste wat Axure kan is het ontwikkelen van documentatie. De mockups worden voorzien van documentatie, en wanneer men klaar is met ontwerpen van het systeem kan er documentatie gegenereerd worden. Het resultaat is een functioneel ontwerp met alle onderwerpen erin.



Figuur 11 - Why Axure?

7.3.1 Matrix

De voor- en nadelen die zijn beschreven zijn te vinden in de bijlage E software oplossing (oplossingsplan). In dit verslag is alleen de matrix opgenomen die is opgesteld in het oplossingsplan. De waardering die gegeven zijn is: ++, +, - en -- waarbij ++ de hoogste is en - de laagste. Er kan afgelezen worden dat Axure enkele positieve effecten heeft op de situatie bij Infoland. De + illustreert dat Axure een positieve bijdrage zal leveren maar de oorzaak niet geheel automatisch zal wegnemen. Ook is te zien dat geen knelpunt geheel wordt opgelost door deze automatisering. Zie Tabel 7 - Axure ramen op Infoland.

Tabel 7 - Axure ramen op Infoland

↓ Problemen	Oorzaken →	Prioriteit (1 hoog / 4 laag)	(o1) Oorzaak 1	(o2) Oorzaak 2	(o3) Oorzaak 3	(o4) Oorzaak 4
P1 Verschillende versies van FO		1	--	++	++	++
P2 Wijzigingen FO niet direct zichtbaar		2	+	+		
P3 FO bestand is groot		3	-			
P4 Aanpassen FO kost veel tijd/werk		1	+	--		
P5 FO niet compleet		3	+	+	--	

7.3.2 Implementatie

Als er gekeken wordt wat er veranderd als Infoland over zou gaan tot het gebruik van Axure dan zou alle huidige functionele ontwerpen overgebracht moeten worden in Axure. Axure biedt geen functionaliteit aan om MS Visio ontwerpen te importeren en dit zou dan handmatig moeten gebeuren per scherm. De schermen over brengen naar Axure met daarbij de documentatie zou een heel langdurig implementatieproject zijn. Men kan er ook voor kiezen om Axure pas te gebruiken bij nieuwe ontwikkelingen en de huidige situatie zo te laten zoals het is. Hier staat wel tegenover dat de architecten de huidige werkwijze zullen moeten opgeven en zich van de vrijheid die ze hebben in MS Visio en MS Word moeten loslaten. Namelijk is het zo dat ze direct in Axure zullen werken en Axure heeft enkele beperkingen bijvoorbeeld ten aanzien van opmaak zoals gewend in MS Word.

7.4 Oplossingen

Tijdens het nader onderzoek zijn er oplossingsideeën bedacht. Niet alle oplossingsideeën zijn bruikbaar voor Infoland en sommige zijn inmiddels al doorgevoerd. In het hoofdstuk van het oplossingsplan wordt er per knelpunt bekeken wat de mogelijke oplossingen zijn. Hieruit volgt een kort advies per knelpunt wat Infoland kan doen om het knelpunt te minimaliseren of te verhelpen.

7.4.1 Verschillende versies van FO (P1)

Dit knelpunt is hoog geprioriteerd in dit project en daarom heeft dit knelpunt ook de meeste aandacht gekregen. Zoals te zien in de matrix van Axure (Tabel 7) lost Axure een groot deel van de oorzaken op bij P1 behalve één oorzaak (o1). Deze oorzaak beschrijft dat erin het proces twee versies van het functioneel ontwerp (FO) ontstaat. De twee versies worden gecreëerd om stilval van het proces te voorkomen. Oplossingsideeën om over te gaan naar een andere manier van werken door bijvoorbeeld een Agile achtige methode toe te passen is niet geschikt voor Infoland, dit hebben ze zelf in het verleden ook onderzocht. Dit knelpunt kan alleen in geheel opgelost worden door procesmatige veranderingen.

Een alternatief om de huidige situatie te optimaliseren is door de software die nu aanwezig is beter te ondersteunen. Voorbeeld ervan kan zijn het gebruik van collaboration software om mogelijk te maken dat de architecten gelijktijdig aan het zelfde document kunnen werken.

7.4.2 Wijzigingen in FO niet direct zichtbaar (P2)

Dit knelpunt beschrijft dat het soms niet duidelijk is voor de ontwikkelaar of tester wat er is aangepast in het functioneel ontwerp na de overdracht. Dit komt omdat het functioneel ontwerp na de overdracht nog kan wijzigen omdat er een fout is ontdekt. Hiervoor is wel een proces ingericht in het systeem genaamd iTask. iTask biedt de mogelijkheid om een formulier op te stellen die op zijn buurt een proces in gang brengt. Binnen Infoland wordt dit gebruikt om ToDo's in te leggen voor onderdelen die aangepast moeten worden in het functioneel ontwerp na de overdracht. Dit ToDo formulier wordt nog niet zo heel lang toegepast en wordt ook nog niet consequent gebruikt. Als er naar het formulier gekeken wordt bevat het maar één tekst box waar alle informatie in verwerkt dient te worden. Hier moet de ontwikkelaar zelf een omschrijving aangeven en kan zelf bepalen welke gegevens volgens hem belangrijk zijn voor deze ToDo. Het ToDo formulier zou verbeterd kunnen worden zodat het meer leidraad zal geven bij het invullen van het formulier. Denk aan een invoerveld voor de schermnummers, oorzaak en oplossing. Op deze manier zal de ToDo ook beter begrepen worden door testers en eventueel andere betrokkenen, zodat het duidelijk is wat er aangepast is in het functioneel ontwerp na de overdracht.

7.4.3 FO bestand is groot (P3)

Dit knelpunt is vrij uniek en leidt tot problemen die vaak van technische aard zijn. Er is gedacht in vele richtingen om dit knelpunt op te lossen. Zo zou bijvoorbeeld de mockups zoals ze gemaakt worden in het functioneel ontwerp kunnen verdwijnen op plaatsen waar een tekstuele beschrijving ook zou kunnen, of de mockups meer veranderen in wireframes zodat ze in minder detail worden getekend. Maar zoals in het diepteonderzoek naar boven komt moet het functioneel ontwerp duidelijk zijn. Het veranderen van de mockups kan resulteren in onduidelijkheid en tevens is het creëren van de mockup voor de architect en manier om zijn werk in te beelden. Het functioneel ontwerp weerspiegelt het product van Infoland en evalueert met de tijd. Hier is verder niet echt een directe oplossing voor dan het advies om het functioneel ontwerp goed te bewaren.

7.4.4 Aanpassen FO kost veel tijd/werk (P4)

Het bewerken van het functioneel ontwerp kost veel werk bijvoorbeeld doordat de mockups gecreëerd worden in MS Visio en altijd gekopieerd moeten worden naar MS Word. Wanneer er iets gewijzigd moet worden dient men de MS Visio tekening te zoeken, bewerken en te vervangen in het MS Word document. Bij dit knelpunt was al snel duidelijk dat er gekeken moest worden naar een softwarematige oplossing. Als er gekeken wordt naar een software oplossing zijn er vaak drie opties: zelf maken, out-of-the-box oplossing of een bestaand systeem aanpassen. Er is gekozen om eerst te kijken of er een out-of-the-box oplossing beschikbaar is in de markt er hiervoor is Axure gevonden die dit knelpunt bewijze van zal wegnemen. Alleen bestaat er nog altijd verschillende versies van het functioneel ontwerp en ook het implementeren van Axure zal heel veel tijd kosten. Hierdoor zal voor Infoland twee opties overblijven. Het is te adviseren om de ontwikkelingen van Axure in de gaten te houden. Ook kan Axure dienen als rol voorbeeld van hoe het anders kan.



7.4.5 FO niet compleet of er ontbreken verwijzingen (P5)

Het ontbreken van onderdelen in het functioneel ontwerp kan voornamelijk gewijd worden aan menselijke fouten. Dit knelpunt is lager geprioriteerd in dit project en heeft daardoor minder aandacht gehad. Dit knelpunt zal immers verdwijnen als men overweegt over te gaan naar een ander systeem of werkwijze. De oplossingsideeën die zijn opgedragen tijdens het nader onderzoek zijn voor een deel al toegepast binnen Infoland. Maar wat nog niet is ingevoerd binnen Infoland is een checklist die dit knelpunt zou kunnen verminderen. Een checklist kan er voor zorgen dat onderdelen die op de checklist staan worden nagekeken. Meeste fouten worden er nu al uit gehaald door te reviewen van het functioneel ontwerp maar er zijn nog steeds zaken die steeds er doorheen gaan. Voor zulke onderdelen zou een checklist opgesteld kunnen worden ter ondersteuning voor het nakijkproces. In combinatie met de gegevens uit de ToDo's kan de checklist aangevuld worden met nieuwe informatie. Want een ToDo zal ingediend moeten worden als er iets ontbreekt in het functioneel ontwerp na de overdracht. Periodiek zou dan de checklist aangepast kunnen worden met die nieuwe gegevens uit de ToDo's.



8 Conclusies en aanbevelingen

Dit project is gestart omdat Infoland problemen heeft met het onderhouden en beheren van het functioneel ontwerp. In een gesprek met de opdrachtgever is de opdracht bepaald, en zijn de volgende vragen opgesteld die beantwoord moeten worden in dit project:

- Welke problemen/knelpunten doen zich voor in het huidige software ontwikkelproces ten aanzien van het ontwerpdocument (functioneel ontwerp) en hoe op te lossen?
- Hoe beheert en onderhoudt men een ontwerpdocument (functioneel ontwerp) op de juiste manier?
- Is er software voor het beheren/fabriceren van een ontwerpdocument (functioneel ontwerp), hoe werkt deze, welk is geschikt voor Infoland en hoe kan die dan het beste ingericht worden voor Infoland?

Om deels antwoord te geven op de eerste vraag is er in het voor- en nader onderzoek interviews gehouden met vijf betrokkenen. Uit die interviews is een lijst met knelpunten opgesteld. Aan de hand van die informatie is er diepgaande onderzoek gedaan om verdere vragen te beantwoorden. Om vraag 2 te beantwoorden is er gekeken wat een functioneel ontwerp is en waar het voor dient. Het doel van een functioneel ontwerp is erop gericht om duidelijk te maken wat er gemaakt moet worden. Maar uit onderzoek blijkt dat de inhoud van het functioneel ontwerp per project kan verschillen. Vandaar dat er geen vaste manier is om het functioneel ontwerp te beheren of te onderhouden. Om de laatste vraag te beantwoorden is erin de markt gekeken naar mogelijke software oplossingen. Om tot de meeste geschikte out-of-the-box softwarepakket te komen is er een pakketselectie uitgevoerd. De uitkomst van deze pakketselectie is een softwarepakket genaamd Axure. Maar uit onderzoek is gebleken dat Axure niet voldoende geraamd kan worden op de huidige situatie bij Infoland, waardoor er geconcludeerd kan worden dat Axure op korte termijn niet zal worden geïmplementeerd.

Uit de resultaten van dit onderzoek kunnen de volgende aanbevelingen worden gedaan om het beheer en onderhoud binnen Infoland te optimaliseren:

- Onderzoek doen naar mogelijke procesmatige veranderingen in het gehele software ontwikkelproces ter verandering van de verschillende versies van het functioneel ontwerp;
- Onderzoek doen om de bestaande software te optimaliseren of te overwegen om een geheel nieuwe software toepassing te ontwikkelen;
- Onderzoek doen na het verbeteren van het ToDo formulier zodat belangrijke gegevens bij het invullen niet over het hoofd worden gezien;
- Onderzoek doen naar het opstellen van een checklist ter ondersteuning bij het nakijken van het functioneel ontwerp;
- De ontwikkelingen van Axure in de toekomst in de gaten te houden.

Evaluatie

Dit is het deel van het afstudeerverslag dat ik terug kijk op mijn functioneren in een persoonlijk evaluatie. Het is een hele leuke periode geweest en begon allemaal op 25 oktober 2011 toen ik mijn sollicitatiebrief en CV opstuurde na Infoland. Vrij kort erna werd ik gevraagd op gesprek. Ik was heel erg gemotiveerd om aan dit project te beginnen en had al vroegtijdig een stageplaats, dus heb goed de tijd gehad om stukken van de PID te schrijven. Vandaar dat ik altijd een goed overzicht had van hoe het project zou gaan verlopen. Toen ik eindelijk februari 2012 aan de slag kon is die motivatie niet veranderd. Door mijn motivatie en enthousiasme heb ik wel leermomenten gehad. Omdat ik ergens in het begin van het project dacht dat ik het 'Ei van Columbus' had gevonden. Dit heb ik uiteraard met vol enthousiasme voorgelegd maar bleek uiteindelijk niet de juiste oplossing te zijn.

Hierdoor is voor mij eigenlijk de opdracht nog duidelijker geworden. Door op deze manier over een oplossing te discussiëren heeft naar mijn mening geleid tot een beter advies. Het gehele traject is goed voorlopen en ben ik methodisch te werk gegaan waardoor ik qua planning geen verdere problemen ondervond. Tegen het einde van het diepteonderzoek kwam voor mezelf wel een tegenvaller. De pakketselectie was uitgevoerd met afgewogen keuzes en daaruit kwam één softwarepakket. Wat in de pakketselectie na boven kwam bleek al dat het pakket net als andere niet geheel ging voldoen aan de gestelde eisen en wensen, maar dit is vaak ook wel te verwachten van een out-of-the-box oplossing. Tijdens het inrichten van de demo (proof-of-concept) en het geven van een presentatie bleek dat de oplossing niet past bij Infoland. Dit vond ik zelf wel echt een tegenvaller omdat ik het probleem echt wou oplossen voor Infoland en tevens ben ik heel perfectionistisch aangelegd. Ik was volgens mij de enige die er negatief tegen aan keek want de rest keek nog altijd positief tegen het resultaat aan. Het heeft daarom ook niet langer dan één nachtje slapen geduurd voor dat ik er overheen was. Ik ben verder gegaan met de onderdelen die nog moesten gebeuren. Zo ben ik ook dieper gaan kijken naar andere oplossingen en mogelijkheden die tijdens het voor- en nader onderzoek na boven waren gekomen. Dit heeft me laten realiseren dat zaken in de praktijk niet altijd mogelijk zijn. Men kan wel de gouden oplossingen bedenken maar het dient wel geaccepteerd te worden en moet passen bij de huidige situatie. Wanneer ik terug kijk op mijn afstudeerperiode bij Infoland kijk ik hier tevreden op terug. Ik ben blij dat ik Infoland in 2011 heb gekozen als mijn afstudeerbedrijf.

Literatuurlijst

1. Infoland, 2012. *Infoland Publieke Website*. [online] toegankelijk op: <<http://infoland.nl>> [geraadpleegd op 20 februari 2012]
2. Infoland, 2012. *Infoland Intranet*. [online] toegankelijk op: <<http://intranet.infoland.local/Default.aspx>> [geraadpleegd op 20 februari 2012]
3. Kempen, P. & Keizer, J., 2011. *Competent Afstuderen En Stagelopen*.
4. Hedeman, B. & Vis van Heemst, G., 2009. *Projectmanagement op basis van PRINCE2*.
5. Vleugel, K., 2009. *Het verbeteringsproces volgens de V2-methode*.
6. Wikipedia, 2011. *Quality function deployment*. [online] (gepubliceerd op: 14 december 2011) toegankelijk op: <http://nl.wikipedia.org/wiki/Quality_function_deployment> [geraadpleegd op 23 maart 2012]
7. Idzenga, K.W.J. & Plu, R.C., 2007. *Wikken en wegen?*. [online] toegankelijk op: <http://www.europeseaanbestedingen.eu/europeseaanbestedingen/download/gunningcriteria/2007_facto_maart_-_wikken_en_wegen__rp_en_ki_.pdf> [geraadpleegd op 23 maart 2012]
8. Indora, 2004. *Software- en leverancier selectie*. [online] toegankelijk op: <http://www.ictaccountancy.nl/downloads/INDORA_Software_en_leveranciersselectie.PDF> [geraadpleegd op 23 maart 2012]
9. Hofland, D., Monschot, K., Hoogstra, J., & Smit, M.C., 2009. *Succesfactoren van pakketselectie en -implementaties*. [online] toegankelijk op: <<http://www.compact.nl/pdf/C-2009-1-Hofland.pdf>> [geraadpleegd op 23 maart 2012]
10. Zijlstra, W., 2006. *Methode en checklist pakketselectie en -implementatie*. [online] (gepubliceerd op: 7 november 2006) toegankelijk op: <<http://zbc.nu/ict/kwaliteitsmanagement-ict/methode-en-checklist-pakketselectie-en-implementatie-deel-1/>> [geraadpleegd op 23 maart 2012]