

Afstudeerverslag

*Een gebruiksvriendelijke manier voor het bewerken
van XML-elementen in FontoXML*

<i>Afstudeerder:</i>	Youri Bosselaar
<i>Studentnummer:</i>	08033153
<i>Bedrijf:</i>	FontoXML (onderdeel van Liones B.V.)
<i>Plaats:</i>	Rijswijk
<i>Examinator:</i>	Chris Heydra
<i>2^e examiner:</i>	Ru Kleijn
<i>Bedrijfsmentor:</i>	Vincent Smedinga
<i>Inleverdatum:</i>	9 januari 2015



FONTOXML

Afstudeerverslag

*Een gebruiksvriendelijke manier voor het bewerken van
XML-elementen in FontoXML*

<i>Afstudeerder:</i>	Youri Bosselaar
<i>Studentnummer:</i>	08033153
<i>Bedrijf:</i>	FontoXML (onderdeel van Liones B.V.)
<i>Plaats:</i>	Rijswijk
<i>Examinator:</i>	Chris Heydra
<i>2^e examiner:</i>	Ru Kleijn
<i>Bedrijfsmentor:</i>	Vincent Smedinga
<i>Inleverdatum:</i>	9 januari 2015

Referaat

In dit document worden de activiteiten die Youri Bosselaar tijdens zijn afstudeerproject heeft uitgevoerd beschreven en geëvalueerd. Dit project werd uitgevoerd in het kader van de opleiding Communication & Multimedia Design, aangeboden door De Haagse Hogeschool.

Trefwoorden

FontoXML

Intuitive design

Liones

Structured content authoring

Usability

User Interface Design

User research

XML authoring

Voorwoord

*“A website without design isn’t a house without art:
it’s a house without a bathroom.”*

- Mike Monteiro -

Dat geldt natuurlijk niet alleen voor websites, maar voor alle digitale producten. Misschien zelfs ook wel voor alle andere producten. Design is deel van het fundament. Design is mede verantwoordelijk voor het falen of slagen van een product. Het gaat niet alleen over hoe het eruit ziet, maar ook over hoe het werkt en voelt. Want al die factoren beïnvloeden hoe gebruikers een product beoordelen. En dat is uiteindelijk cruciaal voor het slagen ervan.

Dit document is geschreven in het kader van de opleiding CMD, die ik volg aan De Haagse Hogeschool. Het document is in eerste plaats gericht aan mijn examinatoren Chris Heydra en Ru Kleijn, maar ook aan mijn bedrijfsmentor Vincent Smedinga. Hen wil ik dan ook graag bedanken voor hun begeleiding, ondersteuning en inzichten tijdens het uitvoeren van mijn afstudeerproject. Daarnaast wil ik collega Bert Willems bedanken voor de brainstormsessies over het grote plaatje van XML-bewerking. Mike Zuidgeest wil ik bedanken voor zijn hulp als tweede observant bij de usabilitytest. Tot slot wil ik alle testdeelnemers van ThiemeMeulenhoff bedanken voor hun tijd, medewerking en het inzicht in hun grijze massa.

Rijswijk, 31 oktober 2014

Youri Bosselaar

Inhoudsopgave

1.	Inleiding	8
	<i>Deel 1: Projectachtergrond</i>	<i>10</i>
2.	Bedrijfsomschrijving	11
3.	Wat is XML?	13
3.1.	Opslagformaat voor documenten	13
3.2.	XML-elementen	13
3.3.	XML-attributen	13
3.4.	XML-schema's	14
3.5.	Waarom XML?	15
4.	Opdrachtomschrijving	16
4.1	Probleemstelling	16
4.2	Doelstelling van de afstudeeropdracht	16
4.3	Resultaat	16
5.	Projectaanpak	17
5.1.	Projectmanagement	17
5.2.	Design	19
5.2.1.	<i>Designbenadering</i>	<i>19</i>
5.2.2.	<i>User Centered Design methode</i>	<i>20</i>
5.3.	Strategie	23
5.3.1.	<i>Projectfaserering</i>	<i>23</i>
5.3.2.	<i>Gekozen activiteiten</i>	<i>24</i>
5.3.3.	<i>Weggelaten activiteiten</i>	<i>26</i>
5.3.4.	<i>Planning</i>	<i>26</i>
	<i>Deel 2: Activiteitenbeschrijving</i>	<i>28</i>
6.	Onderzoeksfase	29
6.1.	Onderzoeken van het product FontoXML	30
6.2.	Onderzoeken van de business objectives van FontoXML	33
6.3.	Onderzoeken bewerking van XML-elementen	35
6.4.	Uitvoeren taakanalyse per bewerking	37
6.5.	Onderzoeken factoren van invloed op XML-bewerking	38
6.5.1.	<i>Onderzoeken van invloed XML-schemaregels op bewerking</i>	<i>38</i>
6.5.2.	<i>Onderzoeken van invloed XML-best practices op bewerking</i>	<i>40</i>

6.5.3.	<i>Prioriteren van bewerkingen voor ontwerp</i>	41
6.6.	Onderzoeken van gebruikers	43
6.6.1.	<i>Vorbereiding van het onderzoek</i>	43
6.6.2.	<i>Uitvoering van het onderzoek</i>	45
6.6.3.	<i>Verwerken van de onderzoeksresultaten</i>	45
6.7.	Analysen vergelijkbare applicaties	47
6.7.1.	<i>Designproblemen selecteren</i>	47
6.7.2.	<i>Kiezen van techniek voor beoordelen</i>	47
6.7.3.	<i>Applicaties selecteren</i>	48
6.7.4.	<i>Beoordelen van de oplossingen</i>	49
6.8.	Opstellen systeemeisen	52
7.	Ontwerpfase	54
7.1.	Opstellen richtlijnen voor ontwerp	55
7.2.	Ontwerpen van interacties rondom bewerkingen	56
7.2.1.	<i>Concepten bedenken</i>	56
7.2.2.	<i>Kiezen van een ontwerptechniek</i>	58
7.2.3.	<i>Uitwerken van de interactieontwerpen</i>	59
8.	Realisatiefase	66
8.1.	Bouwen van high-fidelity prototype	67
8.1.1.	<i>Aanpak bepalen</i>	67
8.1.2.	<i>Bouwen van het prototype</i>	68
9.	Testfase	72
9.1.	Vorbereiden van de usabilitytest	73
9.1.1.	<i>Doelen vastleggen</i>	73
9.1.2.	<i>Testpersonen werven</i>	73
9.1.3.	<i>Methode kiezen</i>	74
9.1.4.	<i>Benodigdheden in kaart brengen</i>	75
9.1.5.	<i>Taken opstellen</i>	75
9.1.6.	<i>Uitvoeren van proeftest</i>	76
9.1.7.	<i>Testplan</i>	77
9.2.	Testen van ontworpen interfacepatronen	78
9.3.	Uitwerken en analyseren van testresultaten	80
10.	Nazorgfase	83
10.1.	Adviseren van de opdrachtgever	84
10.1.1.	<i>Totstandkoming van het advies</i>	84
10.1.2.	<i>Advies op basis van het testrapport</i>	84
10.1.3.	<i>Advies op basis van het onderzoeksrapport</i>	85

11. Procesevaluatie	89
11.1. Behalen van projectdoelen	89
11.2. Algemene projectaanpak	89
11.3. Planning	90
11.4. Taakanalyse	90
11.5. Onderzoeken van Best Practices	91
11.6. Onderzoeken van gebruikers	91
11.7. User Scenario's	92
11.8. Benchmark	92
11.9. Onderzoek naar bestaande interfacepatronen	92
11.10. Voorbereiden van de usabilitytest	93
11.11. Uitvoeren van de usabilitytest	93
12. Productevaluatie	95
12.1. Onderzoeksrapport	95
12.2. Benchmark	95
12.3. Interactieontwerp	95
12.4. Prototype	96
13. Behalen van competenties	97
13.1. Opstellen plan van aanpak	97
13.2. Opstellen interaction design	97
13.3. Uitwerken prototype	97
13.4. Opstellen testplan & testrapportage	98
13.5. Usabilitytest opzetten en uitvoeren	98
Literatuurlijst	99
Verklarende woordenlijst	103

Externe bijlage: Producten van het project

1. Inleiding

Dit verslag is opgesteld naar aanleiding van het afstudeerproject dat ik tussen 1 september 2014 en 9 januari 2015 heb doorlopen bij internetbureau Liones te Rijswijk. Het doel van dit verslag is om inzicht te bieden in het door mij doorlopen project. Om dit te bereiken beschrijf ik de processen, activiteiten en beslissingen waarmee ik bij dit project te maken heb gehad. Op die manier kunnen mijn examinatoren beoordelen of ik mijn project op hbo-niveau heb uitgevoerd. Ook helpt het beschrijven van deze processen en activiteiten mij bij het evalueren van het project. Deze evaluatie is ook opgenomen in dit verslag en beschrijft wat ik heb geleerd, welke goede en slechte punten ik zag na het doorlopen van de activiteiten en hoe ik de kwaliteit van mijn eindproducten waardeer.

Dit document is gestructureerd in drie belangrijke delen: projectachtergrond, activiteitenbeschrijving en evaluatie. Hoofdstuk 2 (het eerste hoofdstuk van deel 1) is de bedrijfsomschrijving. Hierin beschrijf ik kort het bedrijf waar ik mijn project heb uitgevoerd, om een beeld te scheppen voor de lezer. Vervolgens wordt in hoofdstuk 3 beknopt beschreven wat XML precies is. Dit is nodig om de rest van het rapport goed te kunnen lezen. In hoofdstuk 4 komt de opdrachtomschrijving aan bod. Hierin is het achterliggende probleem, de doelstelling en het gewenste resultaat van mijn project te vinden. Het laatste hoofdstuk van de projectachtergrond, hoofdstuk 5, gaat over mijn gekozen projectaanpak. Hier is te lezen welke methoden ik heb gehanteerd bij het uitvoeren van mijn project, en hoe ik die concreet heb toegepast.

Deel 2 van dit verslag is de activiteitenbeschrijving. De hoofdstukken in dit deel zijn ieder gebaseerd op een projectfase. Voor iedere fase vormt een activiteit een paragraaf. In zo'n paragraaf wordt de activiteit beschreven, gemaakte keuzes worden toegelicht en er is een productvoorbeeld te zien. Zo zijn in hoofdstuk 6 de verschillende activiteiten uit de onderzoeksfase beschreven. Hoofdstuk 7 bevat de activiteiten uit de ontwerpfase. Hoofdstuk 8 gaat over de realisatiefase. In hoofdstuk 9 zijn de activiteiten uit de testfase opgenomen en hoofdstuk 10 beschrijft de nazorgfase.

In deel 3 van het rapport is de evaluatie te vinden. Deze is opgesplitst in drie hoofdstukken. Hoofdstuk 11 bevat de procesevaluatie, waarin ik terugkijk op de planning, gekozen activiteiten en de succes- en pijnpunten in de uitvoering daarvan. In hoofdstuk 12 evalueer ik de opgeleverde (tussen)producten uit mijn project. Ik beoordeel de kwaliteit daarvan en mogelijke verbeterpunten voor de toekomst. Tot slot beoordeel ik in hoofdstuk 13 of ik de competenties heb behaald die ik voorafgaand aan het project als doel had opgesteld.

Deel 1

Projectachtergrond

Het eerste deel van dit verslag geeft achtergrondinformatie over het project dat ik heb uitgevoerd. In het eerste hoofdstuk beschrijf ik het bedrijf waar het project is uitgevoerd, zodat de lezer een beeld kan vormen van de omgeving, cultuur en omvang van de organisatie. Daarna leg ik uit wat XML precies is. Deze kennis is nodig om de rest van het verslag beter te kunnen begrijpen. In het derde hoofdstuk beschrijf ik het probleem, doel en gewenste resultaat van mijn opdracht. Het laatste hoofdstuk van dit deel beschrijft mijn aanpak bij dit project. Hier bespreek ik onder andere de gebruikte methoden en technieken.



2. Bedrijfsomschrijving

Om de context van mijn afstudeerproject voor lezers beter inzichtelijk te maken, wordt deze in dit hoofdstuk beknopt beschreven. Het gaat hier zowel om het bedrijf waarbij het project doorlopen is, als om de situatie van het project zoals die was toen ik begon.

Mijn afstudeerproject is uitgevoerd in opdracht van internetbureau Liones B.V., de ontwikkelaar van FontoXML. Liones is een internetbureau uit Rijswijk. Het bedrijf kent ongeveer 35 werknemers, die vooral werken als salesmensen, projectleiders, developers en designers. Liones is in 1999 opgericht door de huidige directeur Jan Benedictus.

De focus heeft in de eerste tien jaar vooral gelegen op het maken van websites voor bedrijven in de uitgeverwereld. Vandaag de dag zijn nog steeds veel uitgevers klant van Liones. Liones heeft voor klant Wolters Kluwer een online-publicatieplatform met bijbehorende XML-editor (de Kluwer Creator) ontwikkeld. Vervolgens ontdekte Liones dat er een markt was voor dit soort producten. Na een korte verkenning van deze markt is Liones eind 2013 begonnen met de ontwikkeling van haar eigen XML-editor: FontoXML.

Deze editor is ook het onderwerp van mijn afstudeerproject. Daarom heb ik dit project in samenwerking met het team van FontoXML (volledig bestaande uit mensen van Liones) uitgevoerd. De mensen uit dit team met wie ik in dit project het meeste te maken heb gehad staan hieronder kort beschreven.



Jan Benedictus

Product Owner & opdrachtgever

Jan heeft als product owner officieel het laatste woord bij beslissingen over FontoXML. In de praktijk worden deze beslissingen echter vaak overlegd met het team, zodat mensen uit alle disciplines hun kennis kunnen gebruiken om te adviseren.



Vincent Smedinga
Lead UX / bedrijfsmentor

Vincent is verantwoordelijk voor de koers en beslissingen op het gebied van User Experience. Hij houdt zich bezig met design en front-end development. Tevens is Vincent mijn bedrijfsmentor bij het afstudeerproject.



Bert Willems
Productarchitect

Bert is productarchitect en bepaalt, samen met Jan en Vincent, de roadmap voor het product FontoXML. Hij is verantwoordelijk voor de implementaties van FontoXML bij meerdere klanten. Daarnaast staat hij regelmatig met Jan op conferenties om FontoXML te promoten. Tot slot fungeert hij in het team als vraagbaak over alles wat met XML te maken heeft.



Youri Bosselaar
Interactieontwerper

Ik stort mij tijdens mijn afstudeeropdracht als interactieontwerper op het ontwerpen van gebruiksvriendelijke XML-bewerkingen in FontoXML. Met de nodige ondersteuning van het team moet de applicatie hiermee gemakkelijker te bedienen worden voor de gebruikers.

3. Wat is XML?

Voor het uitvoeren van dit project, en dus ook voor het kunnen lezen van dit verslag, is het cruciaal om te begrijpen wat XML is. Doordat ik bij aanvang van het project al behoorlijk wat kennis had over dit onderwerp, hoefde ik hiervoor verder weinig onderzoek te doen. Voor de leesbaarheid van dit document beschrijf ik echter alsnog kort wat XML precies is en waarvoor je het kunt gebruiken.

3.1. Opslagformaat voor documenten

XML is de afkorting voor “extensible markup language”. Dit is een coderingstaal die bestaat om *semantiek*(betekenis) en *structuur*(opbouw) van tekstdocumenten te kunnen opslaan. Dat kan natuurlijk ook met tekstverwerkers als Microsoft Word, maar in XML kun je veel meer informatie over je document opslaan. Zo geef je in Word vooral aan dat iets een paragraaf, een titel of een lijst is. In XML gaat dat veel verder. Zo kun je bijvoorbeeld woorden of zinnen markeren als zijnde een persoonsnaam, samenvatting, geboortedatum, vakterm of quote. Een tekstdocument dat wordt opgeslagen in XML, noem je een XML-document. Zo’n document bestaat voornamelijk uit een combinatie verschillende XML-elementen en attributen.

3.2. XML-elementen

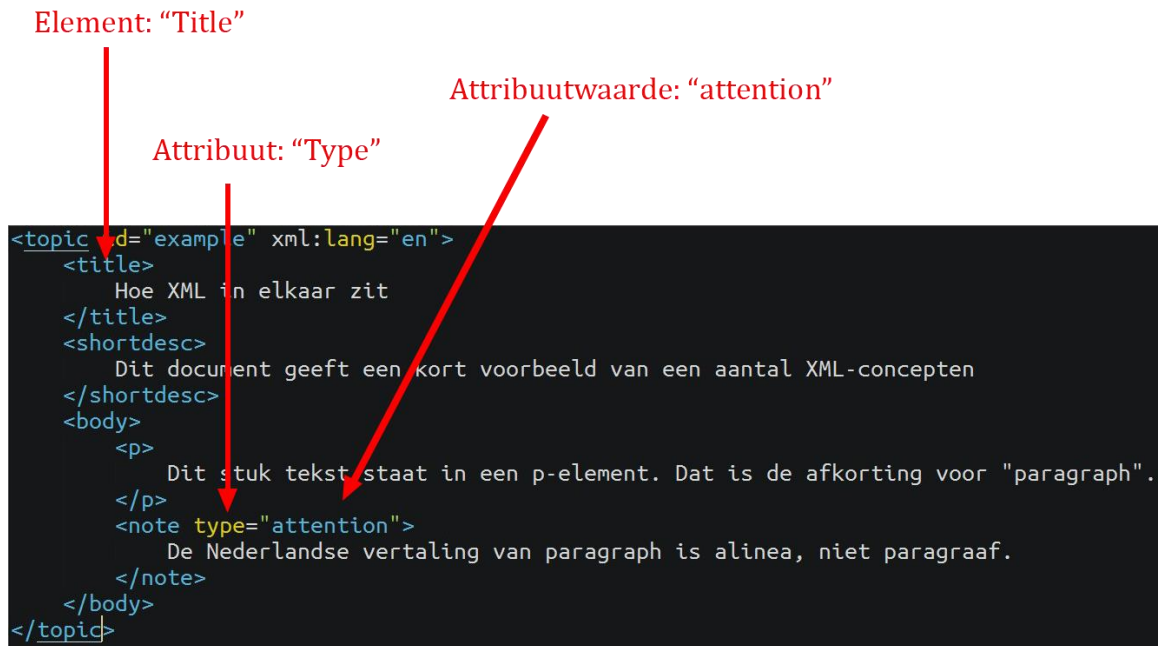
Deze elementen voegen ieder een eigen stukje betekenis aan de tekst toe. Een element kan bijvoorbeeld staan voor een paragraaf, een vakterm, een samenvatting of een auteur van het document. Elementen zijn er dus in allerlei soorten. Het bewerken van XML-elementen is zeer nauw gerelateerd aan het bewerken van tekstdocumenten: XML-elementen zijn alleen maar een manier om de opbouw van die teksten vast te leggen. Het doel van organisaties is dan ook niet per se om te beschikken over XML-documenten. Eerder willen zij beschikken over een manier waarop de structuur en semantiek van hun documenten voor zowel computer als mens inzichtelijk kunnen worden gemaakt. XML is daar de meest voorkomende vorm van.

XML-documenten worden opgeslagen volgens een boomstructuur. Elementen kunnen dus “in” elkaar of “naast” elkaar bestaan. Wanneer ze in elkaar bestaan is er sprake van een ouder-kind relatie. Hierbij is het buitenste element de ouder en het binnenste element het kind. Zo is in afbeelding A het “p”-element een kind van het “body”-element.

3.3. XML-attributen

Een ander veelgebruikt begrip in dit verslag is “XML-attribuut”. Een attribuut kun je zien als een eigenschap van een bepaald element. Het is meestal geen stuk tekst met een bepaalde betekenis; het zegt eerder iets over een element. Zo zou je je voor kunnen stellen dat een notitie-element het

attribuut “type” heeft, waarvan de waarde aangeeft wat voor soort notitie het hier om gaat. In afbeelding A wordt weergegeven hoe een XML-document met daarin elementen en attributen eruit zou kunnen zien.



Afbeelding A. Een XML document met daarin een aantal elementen en een attribuut.

3.4. XML-schema's

Het laatste belangrijke aspect van XML-documenten is dat hun inhoud vaak moet voldoen aan een zogeheten schema. In dit schema zijn regels vastgelegd die gaan over de inhoud van documenten. De meeste en belangrijkste regels gaan over welke elementen er in een document gebruikt mogen worden. Dit kunnen bijvoorbeeld regels zijn over de ouder-kind relatie uit paragraaf 3.2. Een voorbeeld hiervan is de volgende regel: “Een paragraaf-element mag geen hoofdstuk-element bevatten”. Hier is de ouder-kind combinatie “paragraaf > hoofdstuk” dus verboden. Een andere regel zou kunnen zijn: “Ieder hoofdstuk-element moet een titel-element bevatten”. In dit geval is de ouder-kind combinatie “hoofdstuk > titel” dus verplicht.

Zolang een document voldoet aan alle regels uit het achterliggende schema, is er sprake van een schema-valide document. Als een of meerdere van deze regels echter wordt overtreden ontstaat er een *schema-INVALIDE document*. Een XML-document is van weinig waarde als het niet aan het achterliggende schema voldoet. Het is dus van groot belang dat auteurs altijd schema-valide documenten creëren.

Verder bestaan er veel verschillende XML-schema's. Veel bedrijven maken een eigen XML-schema, maar er zijn ook "standaardschema's" beschikbaar. Twee voorbeelden van zulke standaardschema's die veel gebruikt worden door de klanten van FontoXML zijn het TEI-schema en het DITA-schema. Hierbij wordt TEI veel gebruikt voor academische teksten en onderzoeken. DITA wordt veel meer gebruikt voor technische documentatie, zoals handleidingen of lesmateriaal.

3.5. Waarom XML?

Nu heb je een hele lading documenten die zijn opgeslagen in XML. Maar waarom zou je dat eigenlijk willen? XML wordt vooral gebruikt door bedrijven met grote hoeveelheden informatie die verspreid is over een groot aantal bestanden. Er is in dat geval een aantal redenen om XML te gebruiken:

1. *Beter doorzoekbare documenten*

Door documenten op te slaan in XML, worden deze beter doorzoekbaar. Je zou bijvoorbeeld eenvoudig in een database kunnen zoeken naar alle documenten waar de achternaam "Frost" in wordt vermeld. Zo voorkom je dat je ook resultaten krijgt waarin het Engelse woord voor "vorst" is gebruikt.

2. *Gemakkelijk hergebruik van informatie*

Sommige XML-schema's zijn sterk ingericht op hergebruik van informatie. Indien er binnen een bedrijf vaak veel inhoud wordt hergebruikt voor nieuwe versies of publicaties van documenten, kan XML hier van grote waarde zijn.

3. *Breed ondersteund*

XML wordt veel gebruikt in allerlei programma's. Doordat het wordt opgeslagen als eenvoudig tekstdocument is het gemakkelijk uit te lezen en weer te geven.

4. *Snellere multi-platform publicatie*

Doordat de semantiek van documenten veel nauwkeuriger kan worden opgeslagen, kun je voor al deze semantiek aangeven hoe dit eruit moet komen te zien in een publicatie. Dit is voor ieder platform los te bepalen. Misschien wil je een persoonsnaam op een website blauw weergeven en in een boek rood en dikgedrukt. Door de XML hoeven hier niet alle persoonsnamen uit een document te worden veranderd. In plaats daarvan pas je slechts een enkele publicatieregel aan.

4. Opdrachtomschrijving

Mijn opdracht bij dit project was om een groepering van interfacepatronen te vinden die over meerdere schema's herbruikbaar was voor FontoXML. Het omschrijven van de opdracht was het eerste wat ik voor dit project heb gedaan. Omdat inzicht in de opdracht essentieel is voor het begrijpen van de rest van dit verslag, is de gehele opdrachtomschrijving in dit hoofdstuk opgenomen.

4.1 Probleemstelling

Het kostte het team van FontoXML voor aanvang van mijn project veel tijd om voor het bewerken van de verschillende XML-elementen de juiste interface te kiezen. Voor ieder los element moet door het team van FontoXML worden gekeken wat de beste manier van bewerken is en welke interfacepatronen hier bij horen. Mede hierdoor is het implementeren van FontoXML bij nieuwe klanten nog behoorlijk tijdrovend en daardoor duur. Het kost dus nog te veel maatwerk om een interface van FontoXML te configureren.

4.2 Doelstelling van de afstudeeropdracht

Het team van FontoXML wil op een snelle manier de interface van de applicatie kunnen configureren bij een implementatie met een nieuw XML-schema. Daarom moeten er groepen XML-elementen gemaakt worden. Alle elementen binnen zo'n groep moeten met hetzelfde interactiepatroon bewerkt kunnen worden. Het doel is hier om tot een set schema-onafhankelijke groepen te komen. Deze groepen zijn niet zozeer gebaseerd op de elementen zelf, maar op de bewerkingstaken die daarbij horen. Voor elk van deze groepen moet een gebruiksvriendelijk interactiepatroon worden ontworpen. Op deze manier kan de gebruiksvriendelijkheid van de interface van FontoXML gewaarborgd worden zonder dat er voor iedere implementatie een enorme hoeveelheid maatwerk nodig is.

4.3 Resultaat

Na voltooiing van mijn opdracht beschikt het team van FontoXML over een set XML-elementgroepen. Voor minimaal drie van deze groepen is een interactiepatroon ontworpen waarmee de elementen in deze groep gebruiksvriendelijk bewerkt kunnen worden. Dit patroon ligt vast in een interactieontwerp, testrapport en een high-fidelity prototype. Omdat er meerdere patronen ontworpen worden, wordt in een adviesrapport uitgelegd welke het meest geschikt zijn voor gebruik in de applicatie.

5. Projectaanpak

Theoretisch gezien zou je blind aan een project kunnen beginnen en zien wat er op je pad komt. De kans van slagen wordt er dan echter niet groter op. Het is beter om vooraf een strategie voor de aanpak van het project te bepalen. Allereerst moet er een projectmethode gekozen worden.

Hiermee kun je het project plannen en bepalen welke fasen er zullen zijn. Voor een project dat voor een groot deel draait om interactieontwerp, moet er ook een ontwerpmethode gekozen worden. In dit hoofdstuk worden de methoden die ik gekozen heb beschreven en onderbouwd. In paragraaf 5.3 wordt beschreven welke delen van de gekozen methode ik heb toegepast op het project en hoe ik dit gedaan heb.

5.1. Projectmanagement

Bij het kiezen van een projectmanagementmethode is het belangrijk dat deze geschikt is voor het project dat moet worden uitgevoerd. Om dit te kunnen bepalen heb ik een aantal criteria in kaart gebracht die een basis konden vormen voor het kiezen van de meest geschikte projectmanagementmethode:

- *Groot onderzoekscomponent*
Een groot deel van dit project bestaat uit onderzoek. Om zeker te zijn van een compleet resultaat moet dit onderzoek aan een stuk door worden uitgevoerd.
- *Grotendeels alleen werken*
Er wordt binnen mijn project niet veel samengewerkt met een team. De meeste activiteiten worden zelfstandig uitgevoerd.
- *Vaste scope*
De scope van het project zou na het onderzoek duidelijk en definitief kunnen worden vastgelegd. Omdat het ging om een vooraf vastgesteld probleem was het niet mijn verwachting dat de opdrachtgever tussentijds ineens behoefte aan het oplossen van andere problemen zou krijgen.
- *Beperkte beschikbaarheid eindgebruikers*
Er is vooraf aangegeven dat er beschikking is over eindgebruikers, maar slechts beperkt. Het zou dan ook niet wenselijk zijn om bijvoorbeeld elke sprint met eindgebruikers te moeten testen. De voorkeur zou zijn om eenmalig uitgebreid te testen.

De eerste belangrijke keuze die gemaakt moest worden was die tussen een klassieke(waterval-) methode of een Agile methode. Hierbij zijn watervalmethoden vooral geschikt voor grote projecten waarbij er vooraf een duidelijk beeld is van het eindproduct.

Agile is daarentegen meer gericht op iteratief werken, een dynamische lijst met projecteisen en veel contactmomenten tussen team en opdrachtgever. Gebaseerd op de hierboven beschreven punten heb ik gekozen voor een watervalmethode. Er zou namelijk weinig contact met het team nodig zijn en de activiteiten die ik moest gaan uitvoeren waren vooraf duidelijk te bepalen. Vervolgens moest er een specifieke methode worden gekozen die ik kon hanteren bij het plannen van mijn project. Hierbij heb ik gekozen voor de “Projectaanpak in 6 stappen” uit het boek Projectmanagement (Grit, 2011). De voornaamste reden hiervoor was de toegankelijkheid van de methode: het boek van Grit biedt een eenvoudige benadering voor watervalprojecten die universeel toepasbaar is op allerlei projecten. Andere opties, zoals PRINCE2, hebben een grote leercurve en zouden erg tijdrovend zijn. Mede omdat ik dit project zelfstandig moest uitvoeren heb ik ervoor gekozen om dit niet te doen. Het bewaken van het proces en de kwaliteit zou hierdoor te zwaar worden.

5.2. Design

Gebaseerd op het probleem en het doel van mijn project kon worden gesteld dat ik een ontwerpproject moest gaan uitvoeren. Hierbij ging het vooral om het ontwerpen, prototypen en testen van de verschillende interfacepatronen. De methode van Grit biedt vooral een handvat voor het plannen en beheersen van het project in zijn geheel. Om ook in selecteren van de fases en activiteiten een handvat te hebben, heb ik een designaanpak en -methode gekozen. Ook bij deze keuzes was het allereerst belangrijk om een aantal criteria te bepalen waar mijn keuze op gebaseerd moest worden. Hierbij heb ik de volgende criteria opgesteld:

- *Sterke focus op gebruiksvriendelijkheid*
Van www.fontoxml.com: “FontoXML is all about user-friendly authoring of valid XML by occasional authors”. FontoXML hecht erg veel waarde aan gebruiksvriendelijkheid en dit was dan ook zeer belangrijk bij mijn project.
- *Veel nieuwe kennis vergaren*
Het eerste belangrijke punt voor het kiezen van mijn ontwerpmethode was dat ik veel nieuwe kennis moest vergaren. Hierbij ging het zowel om kennis over de gebruikers als om kennis over de context van het probleem uit mijn opdracht. Ik verwachtte dat niet veel van deze kennis uit deskresearch zou kunnen worden vergaard, omdat nog niet veel anderen op dezelfde manier met het probleem waren omgegaan als FontoXML.
- *Aansluiting bij eerdere ervaring*
Dit was geen criterium, maar eerder een pluspunt. Bij twijfel tussen twee methoden op basis van de verdere criteria zou de methode waarmee ik de meeste ervaring had voorkeur krijgen. Dit omdat hiermee kostbare tijd bespaard zou kunnen worden die anders nodig was om de nieuwe methode te leren kennen.

5.2.1. Designbenadering

Voor het uitvoeren van designprojecten kunnen verschillende benaderingen worden gekozen. Een benadering bepaalt de manier waarop designbeslissingen worden genomen en onderbouwd. Ook voor dit project heb ik een benadering gekozen. Bij het kiezen van deze benadering heb ik vooral rekening gehouden met het feit dat er waarschijnlijk maar weinig informatie over het probleem van dit project beschikbaar zou zijn. Hierbij heb ik gekozen uit een lijst van benaderingen die ik zelf samenstelde, gebaseerd op het boek “Designing for interaction” (Saffer, 2010) en het artikel “Five design decision styles” (Spool, 2009). De verschillende benaderingen uit deze lijst worden in deze sub paragraaf kort beschreven.

Self design

Self design is een methode die vooral goed werkt als de ontwerper *zelf* binnen de gebruikersgroep

valt. Op deze manier kunnen er aannames worden gedaan over bepaalde eigenschappen, die dus niet onderzocht zouden hoeven worden. Het kost zo minder moeite om in te beelden hoe de gebruiker iets ziet, omdat de ontwerper zelf een gebruiker zou kunnen zijn. Omdat ik zelf totaal niet binnen het profiel van de gebruikers van FontoXML val, heb ik niet voor deze benadering gekozen.

Genious design

Ondanks het feit dat sommigen genious design niet erkennen als een echte benadering(Nielsen, 2007), wordt deze toch vaak als dusdanig genoemd. Genious design is een benadering die volledig is gebaseerd op de ervaring van de ontwerper. Een ontwerper die in zijn carrière al tien webshops voor schoenen heeft ontworpen, hoeft de elfde keer betrekkelijk weinig onderzoek naar user needs voor te doen. Gezien mijn beperkte ervaring met het onderwerp was dit naar mijn mening dan ook geen geschikte benadering.

Activity Centered Design

Activity Centered Design is volledig gericht op de taken die gebruikers met een product uit moeten voeren. Er wordt hier minder rekening gehouden met de behoeften en eigenschappen van de gebruiker en meer met de benodigde stappen voor het uitvoeren van een activiteit. Ik heb niet voor deze benadering gekozen omdat ik maar weinig kennis over de gebruikers had. Ik ben ervan overtuigd dat deze kennis cruciaal is voor het maken van een gebruiksvriendelijk product.

User Centered Design (UCD)

User Centered Design - ook vaak Human Centered Design genoemd(W3C, 2004) - is een benadering die volledig draait om het vervullen van de behoeften en de doelen van gebruikers. Ik heb voor UCD gekozen vanwege twee belangrijke eigenschappen. Volgens de standaard voor Human Centered Design(ISO, 2010) kan de kwaliteit van een product met UCD sterk worden verbeterd door o.a. de volgende twee punten(vrij vertaald).

1. Het verbeteren van de gebruikerservaring.
2. Het product gemakkelijker te begrijpen en te gebruiken maken en daardoor te besparen op kosten voor training en support.

Zoals genoemd in de criteria, moest er in dit project een sterke focus op gebruiksvriendelijkheid zijn. Daarom waren deze twee eigenschappen van UCD doorslaggevend in mijn keuze: ik heb gekozen om UCD toe te passen binnen dit project.

5.2.2. User Centered Design methode

Zoals gezegd in de vorige sub paragraaf is UCD een benadering voor design. Zo'n benadering bevat bepaalde richtlijnen, concepten en denkwijzen die gehanteerd kunnen worden bij het ontwerpen. Er wordt echter weinig tot geen informatie gegeven over specifieke activiteiten of tussenproducten die kunnen worden toegepast binnen het ontwerpproces.

Wanneer hier wel behoefte aan is, kan een specifieke ontwerpmethode worden gekozen. Ook bij het kiezen van de ontwerpmethode heb ik een aantal criteria gehanteerd. Sommige daarvan zijn hetzelfde als bij het kiezen van de benadering. Andere zijn specifiek gericht op activiteiten en tussenproducten.

- *Veel nieuwe kennis vergaren*
- *Communiceren naar een team met veel ervaring*
- *Snel van concept naar prototype komen*

Ik ben op zoek gegaan naar een passende methode voor dit project, die mij sturing kon bieden in het kiezen van mijn activiteiten en tussenproducten. Ik vond, na enig deskresearch, binnen de benadering van UCD een drietal kandidaten.

LUCID

Het LUCID (Logical User Centered Interaction Design) framework voor het ontwikkelen van interactieve producten. Een van de belangrijkste principes binnen LUCID is die van de vijf E's (effectiveness, efficiency, engagement, error tolerance and ease of learning). Vaak zijn er functionele eisen en gebruikersdoelen binnen een project die met elkaar conflicteren. Met deze vijf E's kan er volgens LUCID balans worden gevonden in de gebruiksvriendelijkheid van een product. Het framework bestaat uit zes fases, waarvan er vier gericht zijn op productdefinitie en design en twee op bouw en implementatie. Omdat ik de laatste twee niet zou toepassen in dit project, heb ik ervoor gekozen om dit LUCID niet te gebruiken. Wel gebruik ik een aantal activiteiten uit de "Discovery"-fase en de "Design Foundation"-fase toe te voegen aan mijn project (Kreitzberg, 2008).

Jesse James Garrett

De methode die binnen mijn opleiding veelvuldig wordt gebruikt, is die van Jesse James Garrett. Ik heb er echter bewust voor gekozen om deze niet te gebruiken bij mijn project. Deze methode is naar mijn mening vooral geschikt voor grote webprojecten zoals het (her)ontwerpen van een website. De focus ligt in deze methode namelijk op twee belangrijke disciplines die daarbij komen kijken: interactieontwerp en informatie-architectuur. Omdat ik verwachtte dat informatie-architectuur in mijn project geen erg groot aandeel zou hebben, heb ik niet voor deze methode gekozen. Wel heb ik uit deze methode het opstellen van systeemeisen als belangrijke activiteit voor het bepalen van de scope meegenomen.

TRUMP

TRUMP is een methode uit 2001 die deels is bekostigd door de Europese Commissie met als doel om de "Human-centredness" van het IT-ontwikkelp proces te verbeteren (Usabilitynet, 2001). TRUMP is een afkorting voor TRial Usability Maturity Process. De methode biedt een lijst van drie basistechnieken en tien aanbevolen technieken die kunnen worden gebruikt in het UCD proces, die zijn gebaseerd op ISO 13407: "*Human-centred design processes for interactive systems*" (nu herzien

naar ISO 9241-210). Op de website van TRUMP wordt per techniek informatie gegeven over het doel, de voordelen en de toepassing hiervan. Ook uit TRUMP heb ik een aantal activiteiten gebruikt, maar niet de gehele methode.

Keuze: hybride

Uiteindelijk heb ik er dus voor gekozen om geen van de hierboven beschreven ontwerpmethoden volledig te gebruiken. In plaats daarvan heb ik een hybride combinatie van activiteiten en tussenproducten samengesteld. Welke dat precies zijn en waarom ik daarvoor heb gekozen wordt beschreven in de volgende paragraaf.

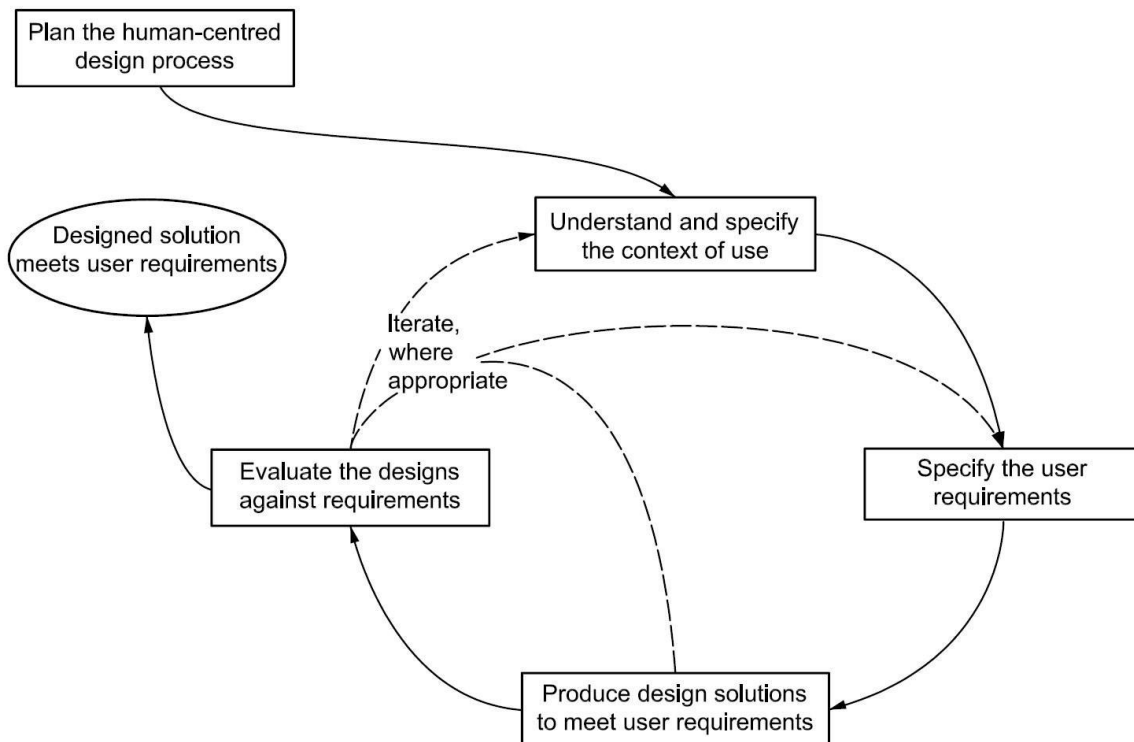
5.3. Strategie

Zoals beschreven in de vorige paragraaf heb ik gekozen voor een projectmanagementmethode, een designbenadering en een set activiteiten uit verschillende ontwerpmethoden. Wat ik uit deze gekozen methode en benadering heb gebruikt en waarom ik voor deze set activiteiten heb gekozen, is te lezen in deze paragraaf.

5.3.1. Projectfaserering

Bij het plannen van dit project heb ik allereerst een fasering gemaakt. Dit heb ik vooral gedaan op basis van de methode van Grit. Zo heb ik, naar voorbeeld uit het boek, een onderzoeksfase, een ontwerpfase, een realisatiefase en een nazorgfase gedefinieerd. Deze heb ik, naar eigen behoefte, nog aangevuld met een testfase omdat dit voor mijn project van groot belang was en ik dit duidelijk als een aparte fase (met een eigen doel) zag.

In de standaard voor User Centered Design worden geen fases beschreven. Wel bevat UCD iets wat ik, bij gebrek aan een betere term, “moederactiviteiten” noem. Dit is een viertal activiteiten(afbeelding B) dat samen de kern van alle UCD projecten vormt.



Afbeelding B. De vier “moederactiviteiten” uit het User Centered Design proces.

Binnen mijn project zijn deze activiteiten in te delen binnen de verschillende fasen van Grit. Zo valt moederactiviteit 1 (Understand and specify the context of use) volledig binnen de onderzoeksfase. Moederactiviteit 2 (specify the user requirements) leeft op de grens van de onderzoeksfase en de ontwerpfase: hier worden de onderzoeksresultaten omgezet in harde eisen voor het ontwerp. Daarna, in de onderzoeksfase, volgt de moederactiviteit "Produce design solutions to meet user requirements". Hier gebeurt het daadwerkelijke ontwerpwerk. De vierde en laatste moederactiviteit is "Evaluate the designs against requirements". Deze valt binnen de door mij gedefinieerde testfase.

5.3.2. Gekozen activiteiten

Het inplannen van activiteiten voor een project begin ik altijd met het stellen van een aantal vragen. Ieder antwoord op zo'n vraag leidt tot een activiteit en uit die activiteit komt een product. Zo'n product kan zo abstract zijn als een lijst met ideeën, maar ook zo concreet als een klikbaar prototype. De vragen die ik bij dit project heb gesteld en de daarbij behorende activiteiten en producten staan hieronder beschreven. Uiteindelijk heb ik alle activiteiten vastgelegd in een strokenplanning, die te vinden is in bijlage B: plan van aanpak.

Business goals

Zoals elk project had dit project bepaalde doelen. Meestal zijn er bij projecten meerdere partijen betrokken, die elk hun eigen belangen en doelen hebben. Ik heb er echter bewust voor gekozen om geen stakeholderanalyse uit te voeren. Door eerdere gesprekken met de opdrachtgever over dit project wist ik dat er, behalve de gebruikers en de opdrachtgever zelf, geen partijen zouden zijn met invloed op het project. Om deze reden was een gesprek met de opdrachtgever en de bedrijfsmentor voldoende voor het bepalen van de projectdoelen. De uitkomst hiervan kon worden vastgelegd in het onderzoeksrapport.

Achtergrond van het probleem

De titel van mijn afstudeeropdracht luidt als volgt: "*Een gebruiksvriendelijke manier voor het bewerken van XML-elementen in FontoXML*". Ik zag het eindproduct hiervan nog niet helemaal voor me. Daarom wilde ik de verschillende onderdelen uit die titel apart gaan onderzoeken en uitwerken. Als eerste: XML, wat is dat nu precies? Vervolgens zou ik onderzoeken wat FontoXML is en wat dat zou betekenen voor dit project. Daarna zou ik onderzoeken wat "*het bewerken van XML-elementen*" precies inhoudt. Hierbij plande ik ook om per bewerking een taakanalyse uit te voeren. Het laatste deel van dit onderzoek moest in kaart brengen welke factoren invloed hebben op het bewerken.

Context van gebruik

In UCD is het begrijpen van de context waarin je product wordt gebruikt een van de onderdelen. Hiervoor heb ik in dit project tweemaal activiteiten gepland.

Allereerst wilde ik leren wie de gebruikers zijn, wat zij precies doen en hoe zij dat doen. Hiervoor heb ik de Contextual Inquiry gekozen, een techniek uit het LUCID-framework. Ik had al vaker gelezen over deze techniek en was van mening dat dit een juiste was voor het achterhalen van de benodigde informatie.

Op basis van de gegevens die ik verzamelde met deze Contextual Inquiries wilde ik twee producten creëren: een lijst met user needs(Garrett) en een aantal scenario's(TRUMP). De eerste wilde ik als basis gebruiken voor een aantal oplossingscriteria. De tweede kon worden gebruikt als kader bij zowel het ontwerpen als het testen van mogelijke oplossingen.

Bestaande oplossingen

Als ik eenmaal zou beschikken over kennis van de context en de probleemachtergrond, kon ik bestaande oplossingen gaan vergelijken. Dit wilde ik op twee manieren gaan doen. Allereerst wilde ik onderzoeken hoe een aantal vergelijkbare applicaties de designproblemen voor dit project oplossen. Dit zou ik vastleggen in een benchmark, met als uitkomst een aantal mogelijke oplossingen. Daarnaast wilde ik onderzoeken of er interfacepatronen beschikbaar waren die deze problemen aanpakken.

Oplossingscriteria

Na het onderzoeken van bestaande oplossingen wilde ik het project gaan afbakenen. Dit hield in dat er criteria moesten worden opgesteld waaraan de oplossingen moesten voldoen. Deze wilde ik vastleggen in systeemeisen, een product dat ik uit de methode van Jesse James Garrett heb meegenomen. Zo zou ik bij het ontwerpen en testen beschikken over een lijst met concrete punten waaraan de verschillende oplossingen moesten voldoen.

Oplossingen ontwikkelen

De volgende stap in mijn project, na het in kaart brengen van de systeemeisen, was het daadwerkelijke ontwerpen. Dit wilde ik gaan doen door middel van schetsen. Deze schetsen zou ik onderbouwen en opnemen in een ontwerprapport.

Evaluatie van oplossingen

User Centered Design draait, zoals de naam al suggereert, volledig om de gebruiker. Een oplossing kan niet worden geaccepteerd als dusdanig voordat deze is getest bij de gebruiker. De gebruiksvriendelijkheidstest was een belangrijke activiteit in mijn project. Om dit te kunnen doen moet er echter een vorm van een prototype zijn. Om te zorgen dat de resultaten hiervan zo veel mogelijk feedback zouden opleveren, heb ik gekozen om een volledig werkend high-fidelity prototype te gaan bouwen. Daarmee konden vervolgens de tests worden uitgevoerd.

Overdracht en advies

De laatste activiteit die ik voor mijn project wilde doen was het overdragen van de gevonden oplossingen. Dit wilde ik doen door, op basis van de testresultaten een aantal adviezen vast te leggen in een adviesrapport.

5.3.3. Weggelaten activiteiten

Er is een enorm aantal activiteiten dat in een UCD project zou kunnen worden uitgevoerd. Ik ga niet voor al deze activiteiten die ik heb weggelaten beargumenteren waarom dat zo is. Wel doe ik dit voor een aantal activiteiten die vaak terugkomt in verschillende methoden, maar niet in dit project.

Stakeholderonderzoek

In een stakeholderonderzoek wordt in kaart gebracht welke partijen invloed op of belang bij een project hebben. Omdat het voor mij al vooraf duidelijk was dat er bij dit (interne) project van FontoXML behalve de opdrachtgever geen partijen betrokken zouden zijn. Om die reden heb ik het stakeholderonderzoek weggelaten uit dit project.

Persona's

In veel UCD projecten worden persona's opgesteld om de eigenschappen van de gebruiker te communiceren binnen het ontwerpteam. Omdat ik dit project volledig zelf zou uitvoeren, heb ik dan ook geen persona's gemaakt.

Paper prototyping

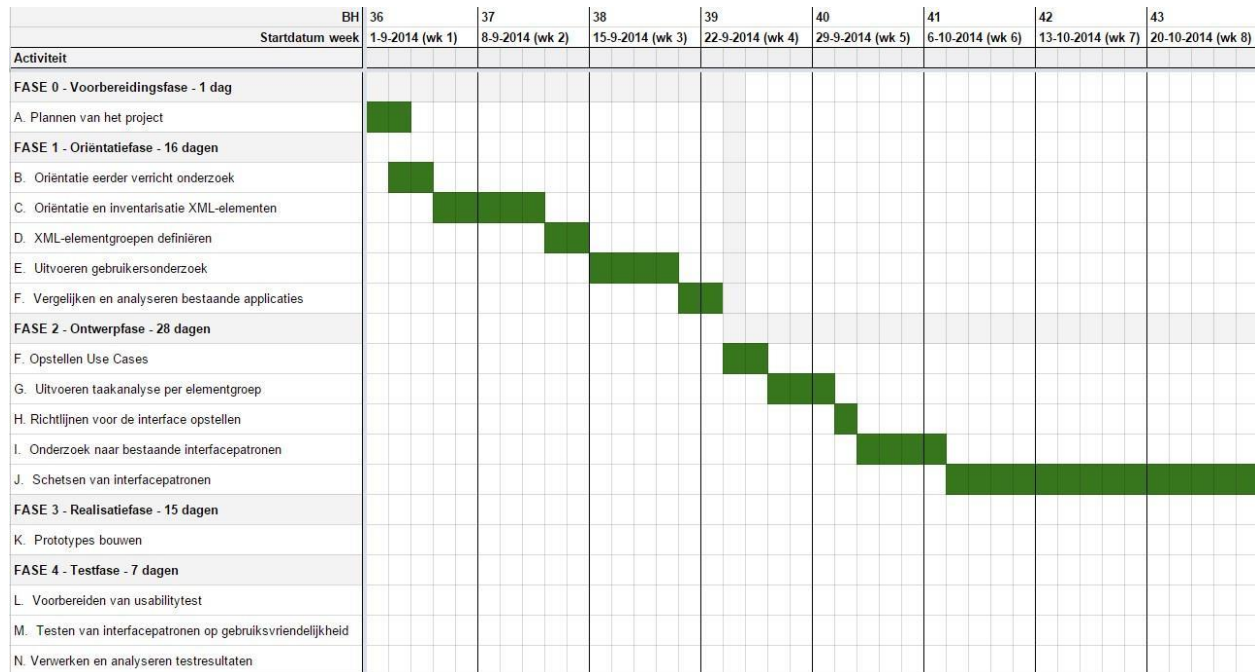
Met papieren prototypes kunnen snel eenvoudige interacties zoals navigatie worden getest. Omdat de interacties in een tekstverwerker echter zeer complex zijn, heb ik geen papieren prototypes gemaakt. Het ontwikkelen hiervan (en daarbij rekening houden met alle mogelijke scenario's) zou een zeer tijdrovend proces zijn geweest. Het leek mij dus beter om de interactiviteit die hier van toepassing is te prototypen in HTML.

Style guide

FontoXML bestaat al een jaar. De stijl is bepaald en vastgelegd. Er was dus geen behoefte aan een door mij ontwikkelde styleguide: het was eerder juist belangrijk dat mijn ontwerpen aansluiten bij de huidige stijl van FontoXML.

5.3.4. Planning

Uiteindelijk heb ik voor alle activiteiten een inschatting gemaakt voor de tijd die zij in beslag zouden nemen. Die inschatting, in combinatie met de volgorde van de activiteiten, heb ik vastgelegd in een strokenplanning. Deze planning is hieronder te zien in afbeelding C.



Afbeelding C-1. De eerste helft van mijn strokenplanning



Afbeelding C-2. De tweede helft van mijn strokenplanning

Deel 2

Activiteitenbeschrijving

In dit deel van het rapport beschrijf ik per projectfase de meest belangrijke activiteiten die ik tijdens het doorlopen van mijn afstudeerproject heb uitgevoerd. Het doel hiervan is om de lezers inzicht te bieden in de activiteiten die ik heb uitgevoerd, hoe ik dat gedaan heb en wat daar mijn motivaties bij waren. Verder is dit ook belangrijk omdat aan de hand van deze activiteitenbeschrijvingen beoordeeld kan worden of de manier waarom ik keuzes maak en onderbouw van voldoende niveau is.

Bij iedere activiteit geef ik ook een voorbeeld van het product dat daaruit voort is gekomen. Deze producten hebben meestal te maken met de ontwerpen voor het designprobleem *“Omgaan met vaste volgorden bij het aanmaken van elementen”* of het designprobleem *“Aangemoedigde element-attribuuat combinaties bij het aanmaken van elementen”*. Op deze manier is de weg van probleem naar oplossing door het verslag heen te volgen. Om onderscheid tussen verslag en voorbeeld duidelijk te maken waar dat nodig is, zijn tekstuele voorbeelden uit (deel)producten weergegeven in een kadertekst.



6. Onderzoeksfase

Het doel van de onderzoeksfase was om te ontdekken welke aspecten in het bewerken van XML-elementen invloed hebben op de interface die voor deze bewerking moet worden aangeboden. Hieruit zou mogelijk een aantal groepen elementen of situaties ontstaan die als basis konden worden gebruikt voor het ontwerpen van gebruiksvriendelijke, herbruikbare interfacepatronen. In de onderzoeksfase heb ik alle informatie verzameld die ik nodig had om mijn producten op een succesvolle, onderbouwde manier te kunnen ontwerpen. Allereerst heb ik onderzocht wat het product FontoXML precies was, om inzicht te verkrijgen in het product waar ik aan ging werken. Vervolgens ben ik gaan onderzoeken wat het betekent om XML te bewerken. Hieruit kwam een lijst met interacties (bewerkingen) die ik kon gaan ontwerpen en een lijst met factoren om rekening mee te houden bij dat ontwerpen. Tot slot heb ik onderzoek gedaan naar de gebruikers van FontoXML. Met de juiste kennis over deze gebruikers kon ik bij het ontwerpen van de interacties rekening houden met hun kennis, ervaring en behoeften. De uitkomsten van alle activiteiten in de onderzoeksfase (behalve het onderzoeken van het product FontoXML) zijn opgenomen in het onderzoeksrapport (bijlage C).



6.1. Onderzoeken van het product FontoXML

De eerste activiteit die ik voor mijn project heb uitgevoerd was het onderzoeken van het product waar het project om draaide: FontoXML. Dit was nodig zodat ik kon begrijpen in welke context mijn ontwerpen toegepast zouden worden. Nog belangrijker was echter om te ontdekken wat de constraints bij het ontwerpen zouden zijn. De ontwerpen moesten namelijk aansluiten bij de productvisie van FontoXML: als de ontwerpen volledig afwijken van deze visie zouden deze immers niet erg bruikbaar zijn. Hierdoor zou de waarde van de ontwerpen voor de opdrachtgever sterk verminderen.

Een groot deel van mijn kennis over het product FontoXML komt voort uit het feit dat ik al bijna een jaar aan het product heb gewerkt voor mijn bijbaan bij Liones. Om deze kennis nog meer toe te spitsen op mijn afstudeerproject, sprak ik met mijn bedrijfsmentor Vincent (Lead UX) en productarchitect Bert. Ik heb hen gevraagd naar het idee achter FontoXML en de concepten waar ik rekening mee zou moeten houden bij het ontwerpen van de verschillende interacties. Een deel van deze informatie had ik wellicht ook van de productwebsite kunnen halen, maar door met mensen uit het team te spreken dacht ik meer diepgaande informatie te kunnen verzamelen. In de kadertekst hieronder is de uitkomst van deze gesprekken te lezen. Deze tekst is afkomstig uit bijlage C: Onderzoeksrapport.

Korte productbeschrijving FontoXML

FontoXML is een webapplicatie voor het bewerken van XML-documenten, gericht op vakexperts met weinig tot geen kennis van XML. In het proces van content creatie wordt veel van de content namelijk geschreven door vakexperts, zoals leraren, monteurs of muzikanten. Zij schrijven deze content dan in Microsoft Word. Omdat veel organisaties momenteel overstappen naar XML als opslagformaat voor hun content, is er vervolgens een conversie nodig van Word-documenten naar XML-documenten. Deze conversie kost veel tijd, moeite en geld. Om te faciliteren in deze nieuwe behoefte wil FontoXML een manier bieden waarop vakexperts hun content direct in XML kunnen schrijven, terwijl het gevoel van schrijven in Word behouden blijft. In de applicatie is er dan ook voor gekozen om het XML-document wat de gebruiker bewerkt niet te tonen als code, maar als het tekstdocument dat die code representeert (afbeelding D). De bedoeling hiervan is om het document beter leesbaar en begrijpelijk te maken, zeker voor mensen met weinig of geen kennis van XML-code.

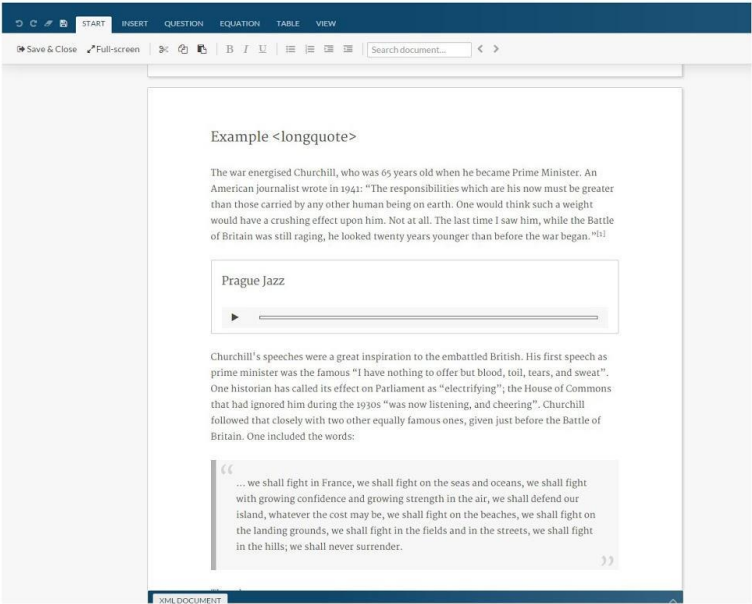




```

<?xml version="1.0" encoding="UTF-8" ?>
<theory id="x" xmlns:xsl="http://www.w3.org/2001/XMLSchema-instance" xsl:
:noNamespaceSchemaLocation="theoryTopic.xsd">
<title>Example <longquote>#62.</title>
<theoryBody>
<p>The war emerged Churchill, who was 65 years old when he became
Prime Minister. An American journalist wrote in 1941: <quot>The
responsibilities which are his now must be greater than those carried by
any other human being on earth. One would think such a weight would have
a crushing effect upon him. Not at all. The last time I saw him, while
the Battle of Britain was still raging, he looked twenty years younger
than before the war began.</quot>H. R. Knickerbocker, <i>Is Tomorrow
Hitler's? 200 Questions on the Battle of Mankind</i>, 1941.</p>
<asset>
<title>Prague Jazz</title>
<audio data="/assets/audio/prague-jazz.mp3">
</asset>
<p>Churchill's speeches were a great inspiration to the embattled
British. His first speech as prime minister was the famous <quot>I have
nothing to offer but blood, toil, tears, and sweat.</quot> One historian
has called its effect on Parliament as <quot>electrifying.</quot> the House of
Commons that had ignored him during the 1930s <quot>was now listening, and
cheering.</quot> Churchill followed that closely with two other equally
famous ones, given just before the Battle of Britain. One included the
words:</p>
<longquote>
<p>... we shall fight in France, we shall fight on the seas and oceans,
we shall fight with growing confidence and growing strength in the air,
we shall defend our island, whatever the cost may be, we shall fight on
the beaches, we shall fight on the landing grounds, we shall fight in the
fields and in the streets, we shall fight in the hills; we shall
never surrender.</p>
</longquote>
<p>The other:</p>
<longquote>
<p>Let us therefore brace ourselves to our duties, and so bear
ourselves, that if the British Empire and its Commonwealth last for a
thousand years, men will still say, <quot>This was their finest hour.</quot></p>
</longquote>
<p>At the height of the Battle of Britain, his bracing survey of the
situation included the memorable line <quot>Never in the field of human
conflict was so much owed by so many to so few.</quot>, which engendered the
enduring nickname <keyword>The Few</keyword> for the RAF fighter pilots
who won it. He first spoke these famous words upon his exit from No. 11
Group's underground bunker at RAF Uxbridge, now known as the <keyword>
Battle of Britain Bunker</keyword> on 16 August 1940.</p>
</theoryBody>
</theory>

```



Afbeelding D. Hetzelfde XML-document wordt getoond in Sublime Text(links) en FontoXML(rechts).

Wat betreft het omgaan met de regels uit XML-schema's heeft het team van FontoXML een aantal belangrijke beslissingen genomen. Omdat documenten die niet aan het bijbehorende XML-schema voldoen, nagenoeg waardeloos zijn, bestaat bijvoorbeeld de volgende richtlijn:

"Het moet niet mogelijk zijn om met de applicatie schema-INVALIDE documenten te creëren."

Een ander idee dat bij FontoXML hoog in het vaandel staat, is het aanmoedigen van de gebruiker om goede content en structuur te creëren. Zo is het bijvoorbeeld gewenst om best practices voor content creatie toe te passen waar mogelijk. De applicatie moet daarbij helpen omdat de gebruikers vaak weinig kennis hebben van XML. Hieruit komt de richtlijn:

"De gebruiker moet, waar mogelijk, worden aangemoedigd in het toepassen van best practices"

Vanwege de complexe aard van XML-documenten, heeft het team ervoor gekozen om de XML-code zoveel mogelijk weg te houden bij de gebruiker. In plaats van deze code te tonen, wordt het document weergegeven als een soort Word-document: een representatie van de semantiek die in de code is vastgelegd. Er wordt dus op een visuele manier geprobeerd om de gebruiker duidelijk te maken hoe het XML-document dat hij bewerkt, is opgebouwd en wat daar allemaal voor

inhoud in zit. Hierbij is het dus belangrijk dat alle (voor de gebruiker relevante) XML-code wordt vertaald naar een visuele vorm.

“Achterliggende XML-code moet zoveel mogelijk bij de gebruiker worden weggehouden. Liever moet de conceptuele betekenis van deze code in het document worden gevisualiseerd.”

Bij aanvang van mijn afstudeerproject is FontoXML ongeveer een jaar in ontwikkeling. Er zijn twee “edities” in de maak, die beide in staat zijn om met een specifiek schema om te gaan. Zo’n editie is een standaardversie van FontoXML waarmee documenten van een bepaald schema kunnen worden bewerkt. Eén daarvan is de “Dita-editie”, de ander de “TEI-editie”. Daarnaast is het team druk bezig met de ontwikkeling van klantspecifieke versies voor onder andere de klanten RILM, IAEA en ThiemeMeulenhoff.

Ondanks het feit dat ik al veel kennis had over FontoXML, leidden deze gesprekken tot een belangrijk resultaat. Ik ontdekte namelijk dat er door het team van FontoXML een aantal richtlijnen was opgesteld (de scheefgedrukte zinnen in de kadertekst hierboven). Deze richtlijnen bleken te zijn bedoeld om de productvisie te bewaken. Dit was voor mij nieuwe informatie: ik wist wel dat dit een deel van de productvisie was, maar niet dat deze richtlijnen ook al concreet geformuleerd waren.

6.2. Onderzoeken van de business objectives van FontoXML

Het is belangrijk om zeker te weten dat mijn projectresultaat de maximale waarde voor mijn opdrachtgever zou bieden. Daarom heb ik in de eerste weken een aantal gesprekken gehad met Product Owner Jan, productarchitect Bert en Lead UX Vincent over wat zij dachten dat het doel van mijn project was. Ik heb deze drie mensen gekozen omdat zij allen een verschillende, belangrijke rol bij FontoXML hebben die van belang is voor mijn project. Deze keuze van mensen heb ik laten bevestigen door aan hen allemaal te vragen wie ik verder nog zou moeten spreken over dit onderwerp. Geen van hen noemde nieuwe namen, waaruit ik concludeerde de juiste personen te pakken te hebben.

Apart van elkaar stelde ik hen allemaal de zelfde vraag: *“Wat hoopt FontoXML te bereiken met het laten uitvoeren van mijn afstudeerproject?”*. Hun antwoorden op de vraag heb ik uitgewerkt in eigen woorden en door hen alle drie laten controleren. Mijn uiteindelijke beschrijving van de business objectives (afkomstig uit [bijlage C: onderzoeksrapport](#)) is te lezen in de kadertekst hieronder.

1. FontoXML wil een gebruiksvriendelijke XML-editor aanbieden.

Het product FontoXML moet zich onderscheiden van andere XML-editors door gebruiksvriendelijk te zijn voor mensen zonder kennis van XML. Een belangrijk onderdeel hiervan is dat FontoXML een interface moet hebben waarmee XML-elementen op een gebruiksvriendelijke manier kunnen worden bewerkt. Doordat het product ontwikkeld wordt aan de hand van de projectmethode SCRUM, is er nog geen grootschalig project gewijd aan de juiste interfacepatronen voor alle mogelijke bewerkingen. Aangezien het wel belangrijk is dat er voor alle bewerkingen een zo gebruiksvriendelijk mogelijk patroon komt, is mijn afstudeerproject in het leven geroepen.

2. FontoXML wil voor nieuwe elementen sneller de interface kunnen samenstellen.

Omdat klanten vaak eigen XML-schema's hebben, heeft het team van FontoXML vaak te maken met nieuwe elementen. Voor hen is het hierbij belangrijk om in zo weinig mogelijk tijd een goed onderbouwde beslissing kan worden genomen over de interfacepatronen voor het bewerken van dit element. Op deze manier kunnen (nieuwe) elementen snel worden opgenomen en bewerkt in de applicatie, zonder dat daarvoor constant denkwerk moet worden gedaan of herhaald. Om hierin te kunnen faciliteren, wordt dit onderzoek uitgevoerd. Concreet hoopt men na het onderzoek te weten:

- in welke bewerkingssituaties bepaalde interfacepatronen kunnen worden gebruikt
- wat de beste positie is voor de trigger voor het aanmaken van ieder element

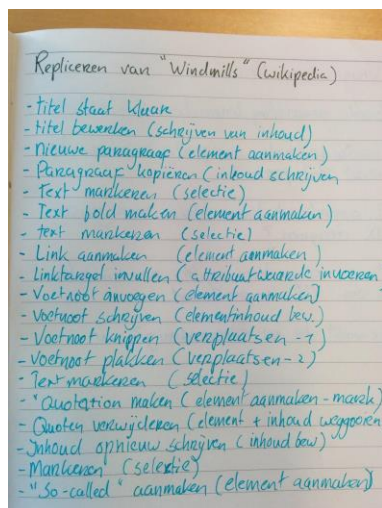
De aanname van de opdrachtgever is dat er op basis van dit onderzoek een beslissingsmodel kan worden ontwikkeld dat helpt bij het kiezen van het juiste interfacepatroon voor een bepaalde bewerking. De conclusie zou echter ook kunnen zijn dat dit model niet bestaat, of niet voldoende dekkend is om meerwaarde te bieden voor het team van FontoXML.

6.3. Onderzoeken bewerking van XML-elementen

Bij dit project was aan mij de opdracht om te ontwerpen hoe je XML-elementen bewerkt in FontoXML. Om dit te kunnen ontwerpen moest ik eerst begrijpen wat “bewerken” dan inhoudt. Ik had daarom behoefte aan een lijst met de mogelijke acties die je kunt uitvoeren bij het “bewerken” van een XML-element, zodat ik deze bewerkingen kon gaan ontwerpen. Ik ben via zoekmachines gaan kijken of zo’n lijst beschikbaar was als basis voor mijn ontwerpen, maar dit bleek niet het geval te zijn. Omdat deze lijst nog niet bestond, was het aan mij om deze lijst op te stellen.

Opstellen van lijst met bewerkingen

Ik heb deze lijst opgesteld zowel in Word als in FontoXML een aantal testdocumenten te bewerken. Hierbij heb ik een rijk DITA-testdocument genomen, dat ik vervolgens in beide applicaties zo nauwkeurig mogelijk ben gaan reproduceren. Op deze manier dacht ik een helder beeld te krijgen van de manier waarop auteurs omgaan met het creëren van content. Door nauwkeurig te noteren welke “bewerkingen” ik constant uitvoerde (afbeelding E.) kwam ik al snel tot een lijst met mogelijke bewerkingen.



Afbeelding E. Aantekeningen met alle acties die ik bij het test-bewerken heb uitgevoerd.

Deze lijst heb ik vervolgens besproken met mijn bedrijfsmentor Vincent en productarchitect Bert. Hierbij heb ik hen gevraagd of deze lijst van bewerkingen compleet (genoeg) was. Na kort te hebben overlegd over mogelijke andere bewerkingen, bleek dat zo te zijn. Ik heb gekozen voor het reproduceren van een bestaand XML-document omdat dit naar mijn idee de meest representatieve manier was om snel een beeld van mogelijke bewerkingen te krijgen. Een alternatief was geweest om echte gebruikers te observeren en te zien welke bewerkingen zij uitvoeren. Nadelen hierbij waren echter dat dit zeer tijdrovend zou zijn geweest. Daarnaast bestond de kans dat gebruikers lang bezig zouden zijn met dezelfde typen bewerkingen.

Per bewerking heb ik ook nog een korte beschrijving gemaakt. Hierin is te lezen wat de bewerking inhoudt en er wordt een voorbeeld van de bewerking gegeven. De volgende beschrijving is voor de bewerking “Element aanmaken”:

“Het is, binnen het schrijven teksten, gebruikelijk om eerst de kale tekst te schrijven en vervolgens deze tekst te selecteren en een bepaalde betekenis toe te kennen. Denk bijvoorbeeld aan het “bold” maken van een stuk tekst in Word. Deze actie bestaat ook als je “niets” selecteert: je cursor staat op een positie zonder selectie.”

De uiteindelijke lijst kent elf bewerkingen en is te zien in het kader hieronder. De uitgewerkte lijst met de beschrijving voor alle bewerkingen is te vinden in [bijlage C: onderzoeksrapport](#).

Lijst van mogelijke bewerkingen op XML-elementen

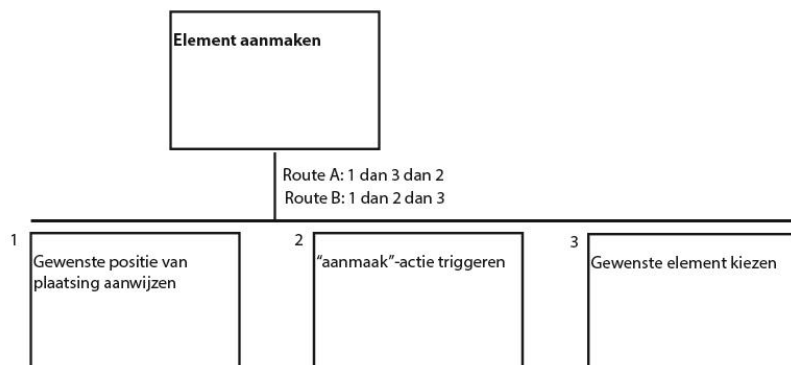
1. Element aanmaken - selectie omwikkelen
2. Element aanmaken - selectie vervangen
3. Element verwijderen - inhoud bewaren
4. Element verwijderen - inclusief inhoud
5. Element verplaatsen
6. Element transformeren naar ander element
7. Element splitsen
8. Element samenvoegen met ander element
9. Attribuut toevoegen
10. Attribuut verwijderen
11. Attribuutwaarde bewerken

6.4. Uitvoeren taakanalyse per bewerking

Nadat ik de lijst met bewerkingen had afgerond heb ik voor iedere bewerking een taakanalyse gemaakt. Deze taakanalyse is een praktisch hulpmiddel bij het ontwerpen van de uiteindelijke interface, omdat hiermee gecontroleerd kan worden of alle (logische) routes van de taak met de interface kunnen worden uitgevoerd. De analyse is dus nodig als controlemiddel voor de compleetheid van een ontwerp. In de oorspronkelijke planning stond deze activiteit pas veel later, in het begin van de ontwerpfase, gepland. Omdat ik hier heel duidelijk bezig was met het in kaart brengen van een aantal taken(bewerkingen), leek de taakanalyse mij hier beter op zijn plaats.

Ook bij het maken van de taakanalyse heb ik de verschillende stappen van een bepaalde actie opgeschreven terwijl ik deze taak uitvoerde. Zo kreeg ik inzicht in welke stappen er allemaal mogelijk waren. Door dit in verschillende (Word en FontoXML) te doen, werden deze stappen al gauw duidelijk. Ik heb bijvoorbeeld in beide applicaties een nieuw “Bold”-element aangemaakt. Hiervoor moest ik eerst met de cursor op een bepaalde positie gaan staan. Vervolgens klik je op een knop waarmee er een “Bold”-element wordt aangemaakt op die positie. Dat leek een enkele stap, maar was eigenlijk een samenvoeging van stap 2 & 3 zoals te zien in [afbeelding F](#).

De resultaten van de analyse staan uitgewerkt in een diagram. In ieder diagram wordt iedere stap die nodig is om een bepaalde bewerking uit te voeren, als blokje weergegeven. Ook is er een beschrijving te lezen van de mogelijke volgorden waarin die stappen kunnen worden uitgevoerd om de taak te voltooien. Ik heb voor deze taakanalyses de HTA(Hierarchical task analysis)-methode gebruikt omdat deze inzicht biedt in de verschillende manieren waarop een taak kan worden uitgevoerd(Hornsby, 2010). Dit was nodig omdat er in het ontwerpen rekening gehouden dient te worden met deze verschillende volgorden: je kunt namelijk niet voorspellen hoe mensen taken uitvoeren.



Afbeelding F. Taakanalyse voor de bewerking "element aanmaken".

6.5. Onderzoeken factoren van invloed op XML-bewerking

Na onderzocht te hebben welke bewerkingen er allemaal mogelijk zijn binnen het bewerken van XML, was er nog niet voldoende informatie om deze bewerkingen te kunnen gaan ontwerpen. Uit het onderzoek naar de productvisie van FontoXML(paragraaf 6.1) was een aantal richtlijnen naar voren gekomen. Deze richtlijnen hebben direct invloed op de bewerking van XML. Om in de ontwerpen om te kunnen gaan met deze richtlijnen heb ik de consequenties die zij hebben op de verschillende bewerkingen onderzocht. Dit bestond uit twee activiteiten, die in de volgende sub paragrafen worden beschreven.

6.5.1. Onderzoeken van invloed XML-schemaregels op bewerking

De eerste richtlijn die naar voren kwam bij het onderzoeken van de productvisie was de volgende:

“Het moet niet mogelijk zijn om met de applicatie schema-invalid documenten te creëren.”

Een XML-schema is dus, zoals eerder beschreven, een document waarin kan worden uitgedrukt aan welke regels XML-documenten moeten voldoen. Er zijn meerdere standaard XML-schema's waaronder DITA en TEI. Verder gebruiken veel organisaties ofwel een eigen schema, ofwel een specialisatie van een standaardschema. Bij zo'n specialisatie worden er bijvoorbeeld nieuwe elementen of regels toegevoegd.

De hierboven genoemde richtlijn wordt, op het moment van dit onderzoek, binnen FontoXML toegepast door te zorgen dat het document op geen enkel moment schema-invalid is. Hier diende rekening mee te worden gehouden bij het ontwerpen van de applicatie. Om te kunnen bepalen wat dit voor invloed heeft op bewerkingen, heb ik allereerst gekeken naar de soorten regels die kunnen worden vastgelegd in XML-schema's. Uiteindelijk vond ik een site waarop alle (veelgebruikte) regels die in schema's kunnen worden uitgedrukt werden beschreven (*Thompson, 2004*). Deze regels heb ik in een lijst gezet zodat ik vervolgens voor iedere regel kon beschrijven op welke bewerkingen deze invloed zou hebben en waarom. Deze lijst bestond uit de volgende regels:

- Toegestane elementen binnen een ouder
- Elementen die verplicht moeten bestaan binnen een ouder
- Elementen die beperkt zijn toegestaan binnen een ouder
- Vaste volgorde van kinderen binnen een bepaalde ouder
- Elementen die elkaar uitsluiten binnen een ouder
- Toegestane element-attribuut relaties
- Vereiste element-attribuut relaties
- Attributen met een verplichte waarde
- Attributen met een beperking in de toegestane waarden

Hierna heb ik voor iedere regel een aantal voorbeelden van elementen opgezocht in de DITA(Eberlein, 2010) en/of TEI(Burnard & Bauman, 2014) documentatie waar deze regels van op toepassing zijn. Ik heb gekozen voor deze twee schema's, omdat deze het meest gebruikt worden door Liones bij implementaties en daardoor leidend waren bij dit onderzoek. Deze voorbeelden zouden zowel helpen bij het begrijpen van de regel als later bij het ontwerpen van de interfacepatronen. De voorbeelden waren enkel bedoeld voor gebruik bij het ontwerpen, wat gedaan wordt door de mensen van FontoXML of mijzelf. Om deze reden was het voldoende om enkel de element- of attribuutnamen te noteren, met daarbij het juiste schema. Zo kon er later worden opgezocht in de documentatie van het schema wat dit verder precies inhield.

Na deze activiteit te hebben uitgevoerd had ik dus voor alle schemaregels een lijst met de bewerkingen waar deze invloed op hebben. Ter illustratie is de uitwerking van de regel "*Vaste volgorde van kinderen*" opgenomen in de kadertekst hieronder. De uitwerking van alle schemaregels is te vinden in [bijlage C: Onderzoeksrapport](#).

Vaste volgorde van kinderen

Er zijn elementen die op een zeer specifieke positie binnen hun ouder moeten staan. Binnen het document kan het dus erg moeilijk zijn om hier met de cursor precies de juiste positie aan te klikken en vervolgens het element te plaatsen. Daarnaast moet de gebruiker een volgorde uit zijn hoofd kennen om de juiste locatie te kunnen vinden. Om de gebruiker te helpen, moet de applicatie vergevingsgezind zijn in dit soort situaties en helpen bij het kiezen van de juiste locatie voor een element.

Factor heeft invloed op:

- Element aanmaken
- Element verplaatsen
- Element transformeren

Voorbeelden:

<taskbody>, <messagepanel>, <step> - (DITA)

6.5.2. Onderzoeken van invloed XML-best practices op bewerking

Een tweede groep factoren die invloed heeft op het bewerken van elementen komt voort uit best practices. In het onderzoek naar het huidige product FontoXML kwam naar voren dat deze best practices in de applicatie zouden moeten worden aangemoedigd waar mogelijk, door de richtlijn:

“De gebruiker moet, waar mogelijk, worden aangemoedigd in het toepassen van best practices”

Ook dit zou invloed kunnen hebben op het bewerken van elementen. Het was voor mij dus van belang om te weten welke best practices er zijn, welke er binnen FontoXML zouden moeten worden aangemoedigd en op welke bewerkingen deze invloed zouden hebben. Met deze kennis zou ik, bij het ontwerpen voor interfacepatronen voor bewerking, rekening kunnen houden.

Om dit te achterhalen ben ik dus allereerst via deskresearch gaan onderzoeken welke best practices er zijn in de bewerking van XML-documenten. Dit bleek een ingewikkelde opdracht, omdat er erg weinig informatie te vinden was over dit onderwerp. De informatie die ik wel vond, was vaak erg technisch en niet gericht op vakexperts zonder XML-kennis. Het zoeken naar deze best practices leidde uiteindelijk tot een aantal dagen uitloop van de planning. Uiteindelijk was het belangrijkste zoekresultaat uit deze activiteit = een document van het W3C (Savourel & Kosek, 2008) waarin een aantal best practices werden beschreven voor *“when authoring XML content”*. Hierin worden negen best practices kort beschreven.

Deze lijst heb ik voorgelegd aan productarchitect Bert, om met hem te bespreken welke wij in FontoXML zouden willen aanmoedigen. Dit bleken echter niet het soort best practices waar het team naar op zoek was. De gevonden lijst bevatte best practices die zeer sterk georiënteerd waren op XML-technische begrippen. Het team wilde zich meer richten op best practices in het creëren van waardevolle content. Omdat ik deze niet kon vinden via deskresearch, besloot ik Bert te vragen of hij een aantal van deze best practices kon noemen. Uit een kort gesprek met hem en Stef (Lead Javascript) kwam ik uiteindelijk tot een lijst van drie best practices:

1. *Aanmoedigen van elementcombinaties*
2. *Aanmoedigen van element-attribuut combinaties*
3. *Bewaken van waardevolle content*

Samen besloten wij dat deze drie het meest belangrijk waren om te behandelen in mijn project. Vervolgens heb ik, net als bij de schemaregels, voor alle best practices een korte beschrijvende tekst geschreven en een aantal mogelijke voorbeelden van toepassing bedacht. Deze voorbeelden waren wat moeilijker te achterhalen omdat deze volgens Bert vaak klantafhankelijk zijn. De elementen die ik als voorbeeld heb genomen zijn echter volgens hem goede mogelijkheden. De uitwerking van de best practice *“Aanmoedigen van element-attribuut combinaties”* is te zien in de kadertekst hieronder. De uitwerking van alle best practices is onderdeel van bijlage C: Onderzoeksrapport.

Element-attributcombinaties aanmoedigen

Er kunnen situaties bestaan waarin een attribuutwaarde “aanbevolen” is. Dit betekent dat invoer niet verplicht is, maar wel moet worden aangemoedigd. In dit geval moet invoer de werkzaamheden van gebruikers dus niet tegenhouden, maar mag er wel om gevraagd worden. Echter, in tegenstelling tot bij een verplichte waarde mag het aanbod hier worden afgewezen.

Factor heeft invloed op:

- Element aanmaken
- Attribuutwaarde bewerken

Voorbeelden (mogelijk)

<foreign> + [lang], <ref> + [href], <date> + [when], - (TEI)

6.5.3. Prioriteren van bewerkingen voor ontwerp

Na het inventariseren van zowel de XML-schemaregels als de best practices, beschikte ik over een grote hoeveelheid factoren die van invloed zijn bij het bewerken van XML-elementen. Hierover trok ik mijn conclusies door de combinatie van factoren en bewerkingen overzichtelijk te maken. Voor het team van FontoXML was het nu van belang dat ik zou ontwerpen hoe deze bewerkingen in FontoXML kunnen worden uitgevoerd. Daarbij moest rekening worden gehouden met alle factoren die daar invloed op hebben.

Na het onderzoeken van deze factoren wist ik dat ik niet genoeg tijd zou hebben om al deze bewerkingen te kunnen ontwerpen, prototypen en testen. Er moest dus een prioritering gemaakt worden binnen de bewerkingen en factoren van invloed. Hiervoor wilde ik echter een overzicht hebben van de verbanden tussen de bewerkingen en de factoren van invloed, dat als hulpmiddel kon dienen bij het overleggen met het team. Om die reden heb ik een kruistabel ([afbeelding G.](#)) opgesteld waarin deze verbanden inzichtelijk worden gemaakt. In deze tabel staat elk blauwe vakje voor een situatie die ontworpen dient te worden: een **designprobleem**. Met deze tabel kon snel worden ingeschat voor welke designproblemen er allemaal opgelost moeten worden, maar ook voor welke bewerkingen het ontwerpen het meeste tijd zou kosten.

Invloed op bewerkingen											
	Aanmaken - Wrap	Aanmaken - Replace	Verwijderen - Unwrap	Verwijderen - Delete	Verplaatsen	Transformeren	Splitzen	Samenvoegen	Attribuut toevoegen	Attribuut verwijderen	Attribuutwaarde bewerken
Schemavaliditeit											
Toegestane ouder-kind relaties											
Vereiste ouder-kind relaties											
Aantal toegestane instanties van een kind											
Vaste volgorde van kinderen											
Elkaar uitsluitende kinderen											
Toegestane element-attribuut relaties											
Vereiste element-attribuut relaties											
Verplichte attribuutwaarde											
Toegestane attribuutwaarde											
Best practices											
Elementcombinaties aanmoedigen											
Element-attribuutcombinaties aangemoedigen											
Bewaken van waardevolle content											

Afbeelding G. Kruistabel waarin de verbanden tussen bewerkingen en factoren van invloed te zien zijn.

Met deze tabel bij de hand heb ik met Vincent overlegd welke bewerkingen hij het liefst ontworpen zou zien. Hierbij was het belangrijk hoe vaak een bewerking voorkomt bij het werken met FontoXML. Onze gezamenlijke inschatting was dat het aanmaken van elementen veruit het meest voor zou komen. Om die reden kreeg de bewerking “aanmaken” prioriteit nummer 1. Daarna werd de bewerking “verwijderen” als meest belangrijke gewaardeerd. De prioritering zoals die uiteindelijk is geworden is te zien in de tabel. Hierbij is de meest linker kolom het meest belangrijk en de meest rechter kolom het minst belangrijk. Deze beslissing is dus vastgelegd in de tabel en verder tekstueel onderbouwd in de conclusies van het onderzoeksrapport(bijlage C).

Het onderzoeksrapport, met deze tabel als belangrijkste uitkomst, was niet alleen een deelproduct, maar ook een van de eindproducten van mijn project. Het was vooraf namelijk al zeer waarschijnlijk dat ik niet alle designproblemen die ik zou ontdekken, ook op kon lossen. Hierdoor ontstond de behoefte aan een duidelijk overzicht van problemen zodat deze na afloop van mijn afstudeerproject alsnog opgelost zouden kunnen worden.

6.6. Onderzoeken van gebruikers

Volgens de richtlijnen van User Centered Design (*ISO, 2010*) is het belangrijk om te beschikken over kennis over gebruiken en behoeften van gebruikers, zodat hier rekening mee kan worden gehouden in het ontwerpen. Om deze informatie te verzamelen heb ik een gebruikersonderzoek uitgevoerd. Dit was al eens gedaan door een voorgaande afstudeerder bij FontoXML, maar dat was bij een andere klant (het IAEA uit Oostenrijk). Omdat deze informatie mij niet volledig vertelde hoe gebruikers de te ontwerpen taken uit mijn project uitvoeren, heb ik ervoor gekozen om bij klant ThiemeMeulenhoff ook een gebruikersonderzoek te houden. Dit stond voor FontoXML sowieso al op de planning, waardoor ik gemakkelijk kon aanhaken. Ik hoefde dus niet zelf op zoek naar testpersonen, omdat ik mijn onderzoek kon combineren met de tests die al gepland stonden. Mijn onderzoek was dus bedoeld om informatie te verzamelen over de mensen die FontoXML nu of in de toekomst gebruiken, zodat er in het ontwerp rekening gehouden kan worden met hun eigenschappen en behoeften.

6.6.1. Voorbereiding van het onderzoek

Als voorbereiding op het onderzoek heb ik allereerst de onderzoeksdoelen opgesteld. Hierin staat beschreven welke informatie ik hoopte te verkrijgen met het onderzoek en waarom precies die informatie. Er zijn twee typen informatie waarvan het gebruikelijk is om deze te verzamelen bij een gebruikersonderzoek: demografische gegevens en etnografische gegevens. Etnografische gegevens gaan vooral over het gedrag en de motivatie van mensen (Hall, 2013). Demografische gegevens gaan vooral over de kwantitatieve aspecten, zoals leeftijd, geslacht en nationaliteit.

Voor dit project heb ik mij vooral gericht op het verzamelen van etnografische gegevens over de gebruikers. Dit omdat ik denk dat deze gegevens meer relevante informatie bieden voor het ontwerpen van interactiepatronen. Zo acht ik informatie over de huidige manier waarop gebruikers taken uitvoeren waardevoller dan informatie over hun gemiddelde leeftijd. Daarnaast zijn deze demografische gegevens ook vooral bruikbaar wanneer er een persona moet worden opgesteld: iets dat ik in dit project niet ging doen. Een persona dient namelijk vooral om naar het team een helder en eenduidig beeld van de gebruiker te communiceren. Omdat ik mijn project nagenoeg alleen zou uitvoeren, was dit in mijn geval niet nodig. Daarnaast bestonden er al persona's voor de applicatie FontoXML in zijn geheel.

De vragen die ik heb opgesteld voor het verzamelen van deze gegevens staan hieronder weergegeven. Deze vragen zijn afkomstig uit bijlage D: Gebruikersonderzoek.

- Wat is de huidige kennis van de gebruikers over gestructureerd schrijven?
- In welke context werken de gebruikers met FontoXML?
- Welke behoeften hebben gebruikers bij het gestructureerd schrijven?
- Hoe voeren de gebruikers het werk dat zij met FontoXML gaan doen momenteel uit?

Vervolgens heb ik een onderzoeksmethode gekozen voor het verzamelen van deze gegevens. Ik ben hierbij dus op zoek gegaan naar een methode om etnografische gegevens te verkrijgen. Via het boek “Just Enough Research” (Hall, 2013) kwam ik terecht bij de Contextual Inquiry. Deze techniek is geschikt voor het verzamelen van etnografische gegevens omdat deze bestaat uit zowel het interviewen als het observeren van de gebruiker. Door te zien hoe deze bepaalde taken uitvoert en tegelijkertijd vragen stelt waar nodig, kan goed worden achterhaald hoe en waarom gebruikers bepaalde dingen doen.

Een deel van de antwoorden op de hierboven beschreven vragen wilde ik verkrijgen in het eerste deel van de Contextual Inquiry: het korte interview. Om hier niet geheel onvoorbereid in te stappen heb ik een klein interviewschema opgesteld. Hierbij heb ik bedacht om het interview semigestructureerd te houden: ik had vooraf wel een aantal vragen opgesteld, maar ik had ook een stukje improvisatie ingecalculleerd. Omdat ik nog niet veel wist over de taken en gebruiken van de gebruikers kon het stellen van doorvragen op het juiste moment naar mijn idee ook waardevolle informatie opleveren. Het uiteindelijke interviewschema zag er als volgt uit:

1. Wat voor werk doet u?
2. Kunt u een korte beschrijving van uw dagelijkse taken geven?
3. Doet u *[authoring / invoeren]* nu ook al?
 - Ja → Ga door naar 4
 - Nee → Doet u wel eens ander werk waarbij u teksten moet schrijven?
4. Wat vindt u van dat werk? Ziet u de waarde ervan in?
5. Op welke manier doet u dat werk momenteel? En met welke programma's?
6. Bent u bekend met XML?
7. Wat zijn prettige punten in huidige manier van werken (breed tot specifiek)?
8. Wat zijn frustraties in huidige manier van werken (breed tot specifiek)?
9. Zijn er dingen die naar uw idee ontbreken in de programma's die u gebruikt?

6.6.2. Uitvoering van het onderzoek

Voor het uitvoeren van de Contextual Inquiries ben ik twee keer naar het kantoor van ThiemeMeulenhoff in Amersfoort geweest. Om goed aan te kunnen sluiten bij de planning van het lopende FontoXML project was het handiger om de onderzoeken van de verschillende gebruikersgroepen te verspreiden over twee middagen. Per middag heb ik dus met een aparte groep gebruikers gesproken. In beide gevallen ben ik op dezelfde manier te werk gegaan. Eerst heb ik in een interview van ongeveer twintig minuten wat algemene gegevens over de gebruiker verzameld. Vervolgens heb ik de gebruiker ongeveer een half uur geobserveerd bij het uitvoeren van zijn dagelijkse taken. Hierbij stelde ik, waar nodig, verdiepende vragen en maakte ik aantekeningen bij opmerkelijke handelingen of opmerkingen.

6.6.3. Verwerken van de onderzoeksresultaten

Na afloop heb ik de conclusies uit mijn onderzoek vastgelegd in een rapport. Voor beide gebruikersgroepen heb ik deze resultaten opgedeeld in drie onderdelen:

1. *“Huidige systeem”*, waarin ik beschrijf met welk systeem de gebruikers hun werk momenteel doen.
2. *“Taken en workflow”*, waarin ik beschrijf wat voor FontoXML-gerelateerd werk de gebruikers doen en hoe dat gaat.
3. *“User needs”*, waarin ik de behoeften beschrijf die gebruikers hebben bij het uitvoeren van die taken.

Verder is er voor beide groepen nog een algemene beschrijving waarin onder andere hun kennis van XML aan bod komt. Als voorbeeld is in de kadertekst hieronder de uitwerking voor de gebruikersgroep “auteurs” opgenomen.

Primaire gebruikersgroep: vakexperts

De eerste en meest belangrijke groep gebruikers bij ThiemeMeulenhoff zijn de vakexperts. In het geval van Thieme gaat het hier over docenten, die zo nu en dan bijdragen aan het schrijven van lesstof. Vaak is dit een langdurig proces waarbij docenten gedurende een periode van maanden, soms zelfs jaren, betrokken zijn. Vakexperts krijgen instructies van de contentontwikkelaars bij Thieme. Deze instructies gaan vooral over het formaat van de inhoud die zij moeten schrijven, maar ook over de leerdoelen waarbij deze moeten aansluiten. Deze vakexperts hebben, zoals de naam al suggereert, erg veel vakinhoudelijke kennis. Zij zijn dan ook cruciaal in het proces van content creatie, omdat zij de content daadwerkelijk creëren. Van achterliggende documentstructuren hebben zij echter een stuk minder kennis.

Zij kennen de aard en opbouw van hun eigen stof, maar zijn niet gewend om deze expliciet vast te leggen in een formaat als XML.

Huidig systeem

De vakexperts werken nagenoeg allemaal in Microsoft Word. Alle ondervraagden hebben hiermee minimaal 5 jaar ervaring. Hierin schrijven zij hun content, ontvangen zij commentaren van redacteurs of contentontwikkelaars en hierin brengen zij verbeteringen aan. Vakexperts komen eigenlijk niet in aanraking met andere software zoals Evolution, laat staan met de XML-code zelf. Hier ligt hun interesse niet en vaak wordt dit ook niet gezien als onderdeel van hun werk.

Taken en workflow

Het is de taak van een auteur om content te creëren. Soms gebeurt dit in een enigszins vrije, ongestructureerde vorm en soms krijgen zij hiervoor sjablonen toegestuurd van een contentontwikkelaar. In dat geval is er de opdracht om de content binnen het sjabloon te schrijven, zodat er al een bepaalde vorm van structuur ontstaat. Natuurlijk is dit nog steeds geen XML en zal het Word-sjabloon dus moeten worden omgezet naar XML in een andere applicatie door technisch redacteurs. Wat opvalt, is dat sommige auteurs wel al aangeven van welk “type” een paragraaf of een hoofdstuk is. Zij plaatsen dan bijvoorbeeld een afkorting voor de titel, waarmee dit type wordt aangeduid. Hier zijn zij dus al bewust bezig met het aanbrengen van semantiek: de stap naar het markeren van een paragraaf lijkt hier een kleine te zijn.

User needs

Uit de observaties zijn meerdere punten naar voren gekomen die van invloed zullen zijn op het gebruik van FontoXML. Om deze punten kort en overzichtelijk te houden, zijn deze geformuleerd als “user needs”. Deze dienen te worden meegenomen in de systeemeisen. Daarnaast moeten deze user needs ook in het achterhoofd worden gehouden tijdens het ontwerpen van de interfacepatronen.

- Vakexperts hebben de behoefte om te kunnen zien welke functie een knop in de interface vervult. Hierbij vragen zij specifiek om tooltips met daarin een korte uitleg.
- Vakexperts zijn volledig gewend aan het werken in Word. Hierdoor ontstaat de behoefte om interacties, waar mogelijk, zo te laten werken als in Word. Zo wordt de leercurve voor het gebruik van FontoXML verkleind.
- Vakexperts hebben behoefte aan begeleiding in het structureren van hun content. Momenteel worden hier vaak Word-sjablonen voor gebruikt.

De uitwerking van het onderzoek is terug te vinden in [bijlage D: Gebruikersonderzoek](#).

6.7. Analyseren vergelijkbare applicaties

Om ideeën op te doen voor mogelijke oplossingen van mijn designproblemen heb ik een aantal bestaande applicaties geanalyseerd waarvan het waarschijnlijk was dat die ook met deze problemen omgaan.

6.7.1. Designproblemen selecteren

In de conclusies van het ontwerprapport was besloten dat ik alleen de bewerking “element aanmaken” zou gaan ontwerpen en prototypen. Om die reden heb ik de tien designproblemen die met deze bewerking te maken hebben, als uitgangspunt genomen bij de benchmark. Deze lijst bestond dus uit de volgende problemen, afkomstig uit het onderzoek naar bewerking van XML-elementen:

1. Omgaan met toegestane ouder-kind relaties
2. Omgaan met verplichte ouder-kind relaties
3. Omgaan met beperkt toegestane elementen
4. Omgaan met vaste volgorde van kinderen
5. Omgaan met elkaar uitsluitende kinderen
6. Omgaan met vereiste element-attribuut relaties
7. Omgaan met verplichte attribuutwaarden
8. Omgaan met toegestane attribuutwaarden
9. Omgaan met aangemoedigde elementcombinaties
10. Omgaan met aangemoedigde element-attribuutcombinaties

6.7.2. Kiezen van techniek voor beoordelen

Het beoordelen van de oplossing die deze applicaties bieden, heb ik gedaan aan de hand van een aantal vuistregels voor usability (Nielsen, 1995). Op deze manier kon ik inschatten welke oplossingen wel of niet de potentie hadden om gebruikt te worden in FontoXML. Mijn project was namelijk volledig gericht op gebruiksvriendelijkheid, waardoor deze vuistregels uiterst geschikt waren als maatstaf. Uit de oorspronkelijke set vuistregels (in het Engels “Heuristics”) van Nielsen heb ik er vijf geselecteerd die naar mijn idee het meest belangrijk waren voor het project. In de lijst hieronder staat voor elk van deze vijf Heuristics de reden dat ik deze heb gebruikt bij de benchmark:

1. *“User control and freedom”*
Bij het bewerken van gestructureerde teksten zijn er enorm veel combinaties van elementen mogelijk. Het is dan ook belangrijk dat gebruikers zo veel mogelijk vrijheid hebben in het indelen van hun document.

2. *“Error prevention”*

XML is ingewikkeld. Er kunnen veel fouten worden gemaakt in de bewerking ervan en ieder schema heeft een enorme hoeveelheid verplichtingen en restricties. Waar mogelijk moet worden voorkomen dat gebruikers hier fouten in maken, omdat dit alleen maar frustratie oplevert.

3. *“Help users recognize, diagnose, and recover from errors”*

Om dezelfde reden als bij 2 kan het bijna niet worden voorkomen dat gebruikers zo nu en dan een fout maken. Het is belangrijk dat zij op dat moment begrijpen wat er gebeurd is en wat er eventueel van hen wordt verwacht om dat weer goed te maken.

4. *“Help and documentation”*

Het schrijven van goede, waardevolle XML vereist kennis. Het is in het geval van FontoXML niet realistisch om te verwachten dat de gebruiker al deze kennis uit zijn hoofd weet. Er zal dus geregeld behoefte zijn aan uitleg of instructie: de applicatie dient hier de juiste informatie op het juiste moment aan te bieden.

5. *“Flexibility and efficiency of use”*

Bij FontoXML wil men de gebruiker zoveel mogelijk in de flow van het schrijven houden. Het is dus van belang dat er zo weinig mogelijke omslachtigheden en onderbrekingen in het bewerken van een document zitten.

6.7.3. Applicaties selecteren

Toen moest er een aantal applicaties worden geselecteerd voor in de benchmark. Hierbij was het belangrijk om de applicaties te kiezen waarvan de oplossingen die zij gebruiken het meest geschikt zouden zijn voor gebruik in FontoXML. Hierbij heb ik gekozen voor de applicaties Google Docs, Microsoft Word, OxygenXML en EasyDita. De eerste twee zijn “gewone” tekstverwerkers, de laatste twee zijn XML-editors.

Waarom ik voor deze applicaties gekozen heb? Allereerst Microsoft Word: een van de bestverkopende tekstverwerkers (Amazon). Omdat de gebruikersgroep van FontoXML gewend is hun content te schrijven in Microsoft Word, was dit de belangrijkste maatstaf op het gebied van interactie met tekstverwerking. Een belangrijke concurrent van Word is Google Docs. Ook deze applicatie is meegenomen in de vergelijking. Dit was vooral omdat deze applicatie net als FontoXML in de browser draait en dus vrijwel platformafhankelijk is. De twee hiervoor genoemde applicaties werken echter niet met XML. Het zijn wel tekstverwerkers, maar missen dus een belangrijke eigenschap die FontoXML wel heeft. Om deze reden heb ik ook nog twee XML-editors meegenomen in de benchmark. De eerste daarvan was OxygenXML, de meest populaire XML-editor op de markt. Oxygen is zeer uitgebreid, maar niet gericht op mensen zonder kennis van XML. Voor het kijken naar functionaliteiten was dit dus een goede kandidaat, maar de waardepropositie verschilt met die van FontoXML. De applicatie waarvan de waardepropositie wél het meest op die

van Fonto lijkt, was EasyDita. Dit is dan ook de vierde en laatste applicatie die is meegenomen in de benchmark. Ondanks het feit dat deze applicatie slechts één XML-schema ondersteunt, is deze wel gericht op mensen zonder kennis van XML. EasyDita probeert, net als FontoXML, aan te sluiten bij de interacties die mensen gewend zijn uit het werken in Microsoft Word.

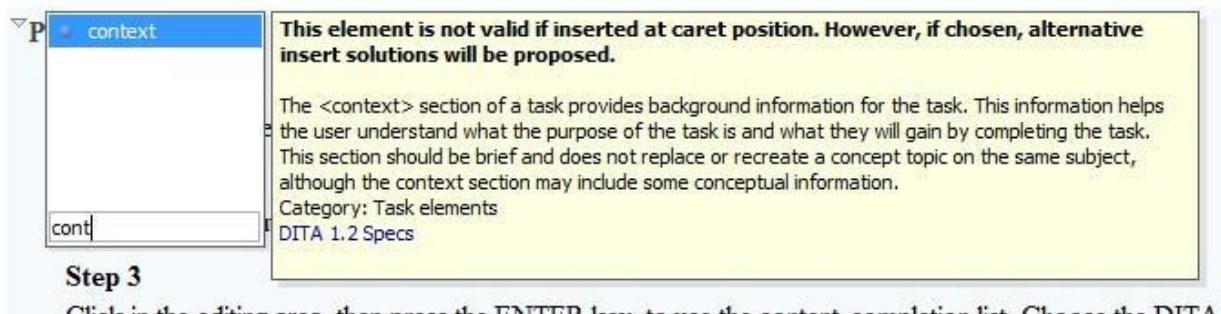
6.7.4. Beoordelen van de oplossingen

Ik ben te werk gegaan door voor elke designuitdaging de voorbeeldelementen uit mijn onderzoek aan te maken in de applicatie. Door dit te doen kon ik testen hoe de vier applicaties met het bijbehorende designprobleem omgingen. De oplossing die ik vond heb ik kort beschreven en waar dat nodig was heb ik ter verduidelijking een screenshot van de situatie opgenomen, zoals bijvoorbeeld bij het probleem “omgaan met vaste volgorde van kinderen” voor OxygenXML, hieronder in de kadertekst.

Omgaan met vaste volgorde van kinderen

Oxygen

Oxygen biedt, bij het toevoegen van een context in een task, de trigger niet als actieve mogelijkheid aan. Je kunt echter wel met de zoekfunctie alsnog proberen een context element aan te maken. Bij focus op die knop geeft een tooltip zowel informatie over het feit dat de trigger inactief is vanwege een invalide positie en over het feit dat er alternatieve acties worden aangeboden. Daarnaast staat er in de tooltip informatie over het element zelf, de categorie waar deze onder hoort en een link naar de DITA specificatie.



De tooltip in Oxygen met daarin verschillende typen verklarende informatie.

Na op enter gedrukt te hebben waarschuwt het systeem je voor de alternatieve actie die uitgevoerd zal gaan worden. Vervolgens moet de gebruiker deze actie bevestigen of annuleren. Voor de technische gebruikers van Oxygen is dit een logische stap: zij willen altijd de mogelijkheid hebben om te bepalen wat er gaat gebeuren. Voor minder technische gebruikers zal het wellicht

een goede oplossing zijn om altijd de meest waarschijnlijke actie uit te voeren, zonder dit verder te vragen.



Oxygen vraagt bevestiging voor het uitvoeren van een alternatieve actie.

Vervolgens heb ik per applicatie de aangeboden oplossing beoordeeld aan de hand van de heuristics. Ik gaf voor elke heuristic een score op de volgende schaal: "--", "-"/+", "+" of "++". Verder gaf ik bij iedere heuristic een korte onderbouwing voor die score. De beoordeling voor het probleem "omgaan met vaste volgorde van kinderen" voor OxygenXML is te zien in tabel hieronder.

Heuristic	Score	Commentaar
User control	+	Er wordt niet zomaar iets anders gedaan dan de gebruiker bedoelt: in plaats daarvan wordt die keuze bij de gebruiker gelegd. Het patroon is hierbij dus behoorlijk vergevingsgezind.
Prevent errors	+	De knop voor het invoegen van context is in eerste instantie uitgeschakeld. Zo wordt foute invoer voorkomen. Zodra deze toch lijkt te ontstaan, biedt het systeem een alternatieve actie aan.
Help users with errors	-/+	Er wordt aangegeven dat er een probleem is (context niet mogelijk op deze positie) en dat er iets anders gaat gebeuren. Wát er gaat gebeuren wordt echter pas in stap 2 uitgelegd. Ook de <i>reden</i> dat het element niet mag worden ingevoegd (vaste volgorde) wordt nergens beschreven.
Help & documentation	+	Bij het invoegen van het element staat er uitgelegd hoe een element gebruikt dient te worden. Op deze manier kan de gebruiker inzicht verkrijgen over
Flexible & efficient	-	Het invoegen van de context is lastig en omslachtig als je niet precies de juiste positie weet. Allereerst is de knop

		uitgeschakeld. Vervolgens moet je ook nog de “alternatieve actie” bevestigen, terwijl die in deze situatie erg logisch is.
--	--	--

Voor veel van de problemen bleek dat Word en Docs geen goed vergelijkingsmateriaal waren: omdat er geen XML bewerkt wordt, hoeven deze applicaties niet met de gedefinieerde designproblemen om te gaan. Wanneer een probleem niet voorkwam in deze applicaties heb ik gezocht naar andere, vergelijkbare situaties om daar nog eventuele ideeën uit te kunnen halen. Dit lukte echter niet voor alle situaties.

Na alle applicaties beoordeeld te hebben op alle designproblemen, heb ik mijn conclusies getrokken. Om als lezer snel te kunnen zien welke oplossingen er zijn opgenomen in de systeemeisen, is hier per designprobleem de oplossing en de applicatie genoemd, zoals hieronder is gedaan voor het *“omgaan met vaste volgorde van kinderen”*.

Omgaan met vaste volgorde van kinderen

Easydita voegt elementen automatisch in op de juiste positie. Dit is een lichte beperking in de gebruikersvrijheid, maar kan een hoop frustratie schelen. Deze oplossing wordt als systeemeis toegevoegd.

In bijlage E is de gehele benchmark te vinden, met daarin de beschrijving en waardering van alle oplossingen en de gehele lijst met conclusies.

6.8. Opstellen systeemeisen

Het onderzoek naar bewerking van XML-elementen leidde tot een grote lijst met designproblemen. Omdat de tijd in dit project maar beperkt is, moest deze lijst worden afgebakend tot een realistisch aantal. Dit heb ik gedaan aan de hand van de tabel met verbanden tussen bewerkingen en invloeden uit paragraaf 5.5 in dit document en de benchmark. Elk blauwe vakje staat daar voor een designprobleem dat kon worden omgezet naar een of meerdere systeemeisen. Voor dit project moest er echter een afbakening worden gemaakt van de designproblemen die ik zou gaan aanpakken. Dit was direct na het onderzoek al gebeurd door te concluderen dat ik enkel de bewerking “element aanmaken” zou gaan ontwerpen. Toen ik de systeemeisen ging prioriteren volgens de MoSCoW-methode(Wikipedia, z.j.) , waren de eisen voor “element aanmaken” dan ook de enigen die de waardering “must have” kregen. Met Vincent had ik afgesproken dat alleen deze bewerking per se binnen mijn project ontworpen en getest moest worden. In de tabel hieronder zijn voorbeelden van deze systeemeisen te zien. De “M” in de rechterkolom staat voor “Must have”.

De gebruiker moet alleen elementen kunnen aanmaken op daarvoor toegestane locaties.	m
Het systeem moet voorkomen dat elementen worden ingevoegd op posities waar zij van het schema niet mogen bestaan door de trigger voor het aanmaken inactief te maken.	m
De gebruiker moet worden geïnformeerd waarom een element op een bepaalde locatie niet is toegestaan.	m
Het systeem moet, bij het aanmaken van een element, eventuele vereiste kinderen automatisch mee invoegen	m
De gebruiker moet, bij het automatisch invoegen van elementen, kunnen zien dat deze zijn ingevoegd en kunnen benoemen wat er gebeurd is.	m
Het systeem moet voorkomen dat een element dat in beperkte mate is toegestaan, vaker dan dat aantal kan worden ingevoegd.	m

De bewerking die een tweede prioriteit kreeg was de bewerking “element verwijderen”. Hierover spraken Vincent en ik af dat deze, indien er tijd over bleek te zijn, ook nog ontworpen zou kunnen worden. Dit was dus geen must, maar een optie. Om die reden zijn de systeemeisen voor “element verwijderen” dan ook geprioriteerd met “could have”. Om te waarborgen dat alle andere gevonden problemen in ieder geval goed waren vastgelegd, zijn al deze problemen alsnog als systeemeis opgenomen. Ook al kon ik deze niet allemaal binnen dit project gaan ontwerpen, dan kon ik (of iemand anders) dit in de toekomst alsnog oppakken. Deze eisen zijn dan ook allemaal geprioriteerd met “won’t have”, omdat zij bewust buiten de scope van het project zijn gelaten. Een aantal voorbeelden van deze eisen is te zien in de tabel hieronder.

Het systeem moet voorkomen dat vereiste kinderen uit hun ouder verwijderd kunnen worden	w
Het systeem moet, waar mogelijk, voorkomen dat het verwijderen van een element onvoorziene invloed heeft op de waarde of validiteit van het document (of een ander document)	w
Het systeem moet de gebruiker informeren wanneer de actie van een gebruiker andere gevolgen heeft dan verwacht.	w

In de systeemeisen heb ik eerst alle bewerkingen als “epics” gedefinieerd. Dit zijn bovenliggende eisen waarvan al gezegd kan worden of deze wel of niet binnen het project vallen. Per epic heb ik vervolgens een aantal aparte systeemeisen opgesteld die meer ingaan op de losse designproblemen. Voor de systeemeisen bij “element aanmaken” zit daar vaak al een oplossing in verwerkt. Deze komt dan voort uit de benchmark. Toch staat ook de systeemeis zonder oplossing nog beschreven, omdat ik ruimte wilde laten voor het ontwerpen van nieuwe oplossingen. In de volgende tabel is een aantal systeemeisen te zien voor onder andere het aanmaken van elementen met een vaste volgorde en het aanmaken van elementen met aangemoedigde attributen. In deze systeemeisen is al een deel van de oplossing verwerkt:

Het systeem moet voorkomen dat elementen worden ingevoegd op posities waar zij van het schema niet mogen bestaan door de trigger voor het aanmaken inactief te maken.	m
Het systeem moet kinderen die op een vaste positie binnen een element moeten staan, automatisch op die positie invoegen. De trigger daarvoor moet, bij focus op de ouder, altijd beschikbaar zijn.	m
Het systeem moet, indien dat qua beschikbare ruimte in het scherm mogelijk is, het invoeren van aanbevolen elementen aanbieden via een popover.	m

Hierbij is de tweede eis een voorbeeld van een eis die gebaseerd is op de conclusies uit de benchmark. Hier bleek dat EasyDita de beste oplossing biedt voor dit probleem. Deze oplossing is dan ook meegenomen als mogelijke oplossing binnen FontoXML. De complete lijst met systeemeisen voor alle designproblemen is te vinden in [bijlage F: Systeemeisen](#).

7. Ontwerpfase

In de ontwerpfase gaat het om het bedenken en uitwerken van oplossingen voor de designproblemen die zijn voortgekomen uit de onderzoeksfase. Sommige van deze oplossingen zijn al gedefinieerd in de systeemeisen, sommige zullen nog bedacht moeten worden. Deze oplossingen worden vastgelegd in de vorm van schetsen. Deze schetsen worden, waar nodig, nog verder onderbouwd in het *ontwerprapport*. Hierin zijn ook de richtlijnen te vinden die ik heb gehanteerd bij het ontwerpen van de oplossingen. Op basis van de schetsen kan in de realisatiefase een aantal prototypes worden ontwikkeld.



7.1. Opstellen richtlijnen voor ontwerp

Voor de interactieontwerpen was het belangrijk dat deze gebruiksvriendelijk zouden zijn en goed zouden aansluiten bij de bestaande applicaties van FontoXML. Om dit te kunnen aanmoedigen en bewaken, heb ik een aantal richtlijnen opgesteld. Omdat de achterliggende gedachte voor deze richtlijnen van toepassing is op de meeste designproblemen, zijn deze niet voor iedere systeemeis apart geformuleerd. Liever zou ik deze richtlijnen bij de hand houden als ik ging ontwerpen. Zo kon ik af en toe checken welke richtlijnen voor een probleem van toepassing waren en of het ontwerp goed aansloot bij deze richtlijnen.

De richtlijnen heb ik gebaseerd op de algemene richtlijnen van FontoXML (zie hoofdstuk 5.1). Daarnaast heb ik een aantal algemene vuistregels voor usability gespecialiseerd voor mijn project. Zo heb ik de vuistregel *“Help users recognize, diagnose, and recover from errors”* (Nielsen, 1995) gebruikt als basis voor de volgende richtlijn:

Verklaar onverwachte situaties en situaties die XML-kennis vereisen

Gebruikers weten niet wat XML inhoudt. Op het moment dat zij dus geen “person name” in een “surname” mogen plaatsen, is dit verwarrend. Als een element niet mag worden verwijderd omdat het verplicht is in zijn ouder, is dit verwarrend. Al deze situaties vereisen duidelijke feedback om frustratie bij de gebruiker weg te nemen. Door begrip voor de situatie te creëren, wordt deze frustratie tot een minimum gereduceerd. Als er schemakennis nodig is voor het correct uitvoeren van een taak, dient deze dus op het juiste moment aan de gebruiker te worden aangeboden.

De richtlijnen die ik bedacht heb ik vervolgens besproken met Vincent. Als verantwoordelijke voor de User Experience van FontoXML vond ik hem de beste persoon om deze richtlijnen mee te bespreken. Na een aantal kleine verbeteringen keurde hij mijn richtlijnen goed. Deze heb ik vervolgens opgenomen in bijlage G: Ontwerprapport.

7.2. Ontwerpen van interacties rondom bewerkingen

Met een complete lijst systeemeisen en een set duidelijke ontwerprichtlijnen, kon ik beginnen met het ontwerpen van de oplossingen voor de designproblemen. Voor de bewerking “element aanmaken”, die ik zou gaan ontwerpen, was dit in totaal een lijst van tien problemen.

1. Omgaan met toegestane ouder-kind relaties
2. Omgaan met verplichte ouder-kind relaties
3. Omgaan met beperkt toegestane elementen
4. Omgaan met vaste volgorde van kinderen
5. Omgaan met elkaar uitsluitende kinderen
6. Omgaan met vereiste element-attribuu relaties
7. Omgaan met verplichte attribuutwaarden
8. Omgaan met toegestane attribuutwaarden
9. Omgaan met aangemoedigde elementcombinaties
10. Omgaan met aangemoedigde element-attribuucombinaties

Samen met Vincent heb ik besloten dat ik voor elk van deze problemen meerdere (minimaal 2) oplossingen zou gaan ontwerpen. Dit was wenselijk omdat deze problemen nog niet erg vaak waren opgelost binnen een productvisie als die van FontoXML. Door meerdere oplossingen te ontwerpen en te testen bleef de kans bestaan dat er een nieuwe, betere oplossing gevonden zou worden voor een van de problemen dan bijvoorbeeld uit de benchmark was gevonden.

7.2.1. Concepten bedenken

Het eerste wat ik per designprobleem heb gedaan is het bedenken van concepten. Hierbij schreef ik om te beginnen het probleem uit dat moest worden opgelost. In deze beschrijving gaf ik aan wat het probleem precies was en in welke richting de oplossing gezocht kon worden. Zo maakte ik voor het designprobleem “*vaste volgorde van kinderen*” de volgende beschrijving:

Vaste volgorde van kinderen

Soms moeten de kinderen van een element in een bepaalde volgorde staan. Omdat triggers (in designprobleem 1) zijn uitgeschakeld op posities waar het element niet mag worden toegevoegd, moet de gebruikers in dit geval dus exact de goede positie aanwijzen om zo’n kind in te voegen. Hierbij doen zich twee problemen voor: ten eerste kan het lastig zijn om de exacte cursorpositie te bereiken waar het element is toegestaan. Ten tweede is het zeer waarschijnlijk dat de gebruiker de exacte volgorde niet uit zijn hoofd weet, waardoor hij überhaupt niet weet waar hij moet zoeken naar de juiste cursorpositie.

Bij dit probleem is het van belang om vergevingsgezind te zijn naar de gebruiker. Help de gebruiker met het vinden van de juiste locatie, of corrigeer de gebruiker bij foutjes. Daarnaast is het ook hier belangrijk om te situatie te verklaren: vertel de gebruiker dat er een vaste volgorde van toepassing is op bepaalde elementen en wellicht zelfs welke volgorde dit dan is. Wees echter niet te opdringerig. Toelichtingen moeten de gebruiker helpen, niet in de weg zitten.

In dit voorbeeld is onder andere te lezen dat gebruikers moeite zullen hebben met het vinden van een toegestane cursorpositie bij elementen met een vaste volgorde. Deze stelling heb ik gebaseerd op het *gebruikersonderzoek*, waarin bleek dat gebruikers meestal geen kennis van XML hebben. Om het probleem snel en sprekend kenbaar te maken bedacht ik ook bij ieder probleem een “gebruikersquote”. Deze baseerde ik op gegevens uit het gebruikersonderzoek of op een van de ontwerprichtlijnen. Voor het designprobleem dat hierboven wordt beschreven heb ik, om het probleem aan te duiden, de volgende gebruikersquote opgesteld:

Gebruikersquote:

“Waarom kan ik niet gewoon een voorbeeld aan deze task toevoegen?”

Deze quote is gebaseerd op de ontwerprichtlijn *“Verklaar situaties onverwachte situaties of situaties die XML-kennis vereisen.”* Door aan te duiden dat de gebruiker niet begrijpt waarom een actie niet nodig is, geef ik aan dat er in de oplossing bepaalde uitleg gegeven moet worden.

Na het probleem te hebben uitgeschreven ging ik op zoek naar inspiratie voor mogelijke oplossingen. Dit deed ik allereerst in de benchmark: hier was immers al een aantal oplossingen beschreven die mogelijk geschikt zouden zijn voor toepassing in FontoXML. Zo vormde de uitkomst van de benchmark voor het designprobleem *“vaste volgorde van kinderen”* de basis voor beide ontwerpen die ik heb gemaakt. In allebei de ontwerpen worden elementen automatisch op de juiste positie ingevoegd, zoals dat ook in EasyDita gebeurt.

In de benchmark was er echter per designprobleem maximaal 1 goede oplossing beschreven. Om deze reden heb ik ook nog verder gezocht naar oplossingen in andere bronnen. Allereerst keek ik hierbij naar designwebsites zoals Dribbble.com, Behance.net en Pinterest.com. Hier zijn veel voorbeelden te vinden van interfaceontwerpen die zeer uiteenlopende problemen oplossen. Een andere bron die ik veel heb gebruikt is het boek *“Designing Interfaces”* (Tidwell, 2011). Hierin staat een groot aantal interfacepatronen beschreven, gesorteerd op het type probleem dat zij oplossen. Door te zoeken naar de problemen die ik in de beschrijving benoem, vond ik hier veel ideeën. Als laatste heb ik ook nog een aantal eigen ideeën gebruikt in de ontwerpen. Deze waren dan gebaseerd op bestaande patronen die ik kende en heel soms zelfs op een nieuw patroon dat ik bedacht. Voor ieder concept dat ik bedacht nam ik de titel vast op in het ontwerprapport. Soms maakte ik hierbij

al wat vroege schetsen om de oplossing beter te kunnen onthouden. De resultaten van deze activiteit (de vroege schetsen) heb ik uiteindelijk niet opgenomen in een (tussen)product. Ze waren enkel bedoeld als opstapje naar nieuwe ideeën.

7.2.2. Kiezen van een ontwerptechniek

Als “eindproduct” in de ontwerpfase heb ik gekozen voor interactieschetsen op papier, die waar nodig tekstueel worden onderbouwd. Ik heb er dus bewust voor gekozen om geen digitale mockups van welke fidelity dan ook te maken. Dit heb ik gedaan vanwege een aantal belangrijke redenen, waaronder de volgende:

Focus op interactie, niet op grafisch ontwerp

In mijn project lag de nadruk sterk op interactieontwerp en maar nauwelijks op grafisch ontwerp. Het was van belang om begrijpelijke, gebruiksvriendelijke interfacepatronen te ontwikkelen waarmee gebruikers bepaalde taken goed konden uitvoeren. Zolang de beperking in grafisch ontwerp de gebruiksvriendelijkheid niet zou beïnvloeden, was dit dus van lage prioriteit.

Design in de browser

Ik zou van schetsen direct naar de browser (HTML/CSS) gaan om mijn prototypes te ontwikkelen. Hier zou de benodigde laag grafisch ontwerp kunnen worden toegevoegd aan mijn ontwerpen. Omdat de huidige huisstijl al uitgebreid was opgezet (ook in de CSS-architectuur), zou deze gemakkelijk toe te passen zijn. Door in mijn schetsen alle interacties al goed uit te werken, zou alle informatie voor het uitwerken van een prototype beschikbaar zijn.

Kwantiteit boven detail (Buxton, 2007)

Omdat er nieuwe ideeën bedacht moesten worden, wilde ik deze snel kunnen uitwerken en bespreken met het team. Slechte ideeën vallen af, goede kunnen doorontwikkeld worden. Door snel te itereren op schetsen wilde ik tot nieuwe concepten komen voor het oplossen van problemen. Vervolgens zou ik deze schetsen uitwerken tot storyboards waarin de interactie beschreven wordt.

Genoeg detail voor conversatie

Ondanks het feit dat een schets maar weinig kleur en vormgeving bevat, biedt deze genoeg detail om over een ontwerp te kunnen sparren met anderen. Door een aantal consistente stijlregels te hanteren blijven de schetsen makkelijk te interpreteren.

Natuurlijk heeft het gebruiken van enkel schetsen t.o.v. digitale mockups ook nadelen. Zo is het gevoel bij het ontwerp binnen de echte applicatie moeilijker op te wekken vanuit een schets. Ook zullen de verhoudingen tussen elementen binnen een schets vaak onrealistisch zijn. Dit kan naar

mijn mening echter gemakkelijk opgelost worden bij het bouwen van het prototype. Hier zijn formaten heel gemakkelijk en proportioneel aan te passen. Dit kost veel minder tijd dan het bij het ontwerpen (zowel schetsen als mockups maken) zou kosten en is dus een goede oplossing. Een ander nadeel is dat schetsen moeilijker “te lezen” zijn dan mockups. Het kan voor klanten veel moeite kosten om een schets te begrijpen en het eindproduct daarbij voor te stellen. Omdat mijn “klanten” echter professionele webdesigners- en developers waren, verwachtte ik hierbij geen problemen.

Een andere goede optie voor het eindproduct was het maken van “digitale schetsen”. Hierbij zou ik in Adobe Illustrator, met behulp van de Interactive Sketching Notation (Linowski, 2012) de verschillende ontwerpen hebben uitgewerkt. Een groot voordeel hiervan is dat er, met behulp van een aantal herbruikbare onderdelen, snel een basis neergezet kan worden. Omdat ik echter ook veel “custom” elementen verwachtte in mijn schetsen, heb ik ervoor gekozen om dit niet te doen. Het maken van deze custom elementen in Illustrator kost naar mijn ervaring namelijk erg veel tijd, die ik beter kon gebruiken voor het produceren van nieuwe schetsen.

Voordat ik ging schetsen heb ik dus eerst nog een aantal stijlregels opgesteld. Deze hanteer ik al langer, maar als checklist heb ik ze toch nog even uitgewerkt. De stijlregels die ik heb gebruikt zijn:

1. *Commentaren in rood.*

Op deze manier is snel te zien wat er bij het ontwerp hoort en wat er commentaar is.

2. *Kleur is voor nadruk, niet voor vormgeving.*

Kleur kan worden toegepast om bepaalde onderdelen te benadrukken. Vormgeving wordt in deze schetsen verder buiten beschouwing gelaten

3. *Gebruik het juiste niveau van detail.*

Ga geen heel scherm schetsen als je slechts een knop in de toolbar wilt uitdrukken. Door op het juiste niveau te werken wordt de aandacht naar de juiste punten gebracht.

4. *Laat duidelijk de stappen van een interactie zien.*

Als ontwerper heb je heel duidelijk in je hoofd hoe iets moet werken. Het doel van de schetsen is echter om dit ook aan anderen kenbaar te maken. Vergeet dus geen stappen te beschrijven, het is vervelend als een ander die moet gaan raden.

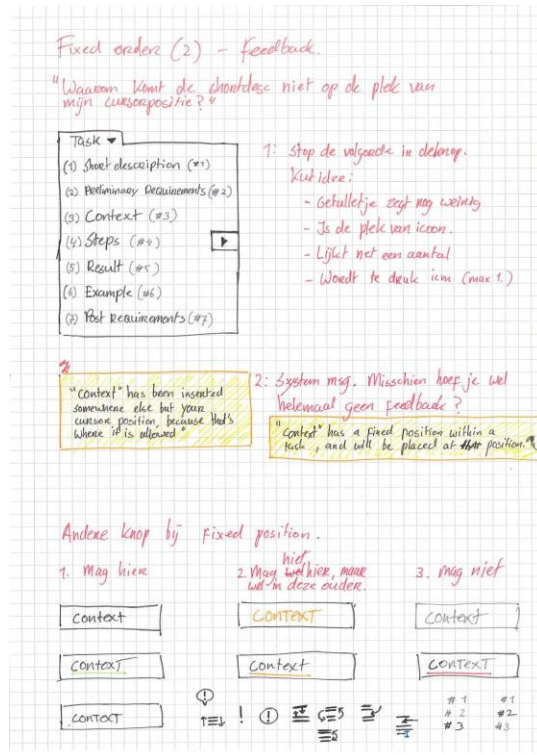
5. *Als je iets niet kunt schetsen, beschrijf het dan in commentaar.*

Het naar rechts sliden van een paneel is moeilijk te schetsen, of je moet een flipbook maken. Als het niet lukt om iets te schetsen, beschrijf het dan tekstueel.

7.2.3. Uitwerken van de interactieontwerpen

Bij het schetsen ben ik dus iteratief te werk gegaan. Eerst werkte ik een concept op verschillende manieren snel en grof uit, zodat ik zelf kon nadenken over de voor- en nadelen van deze oplossingen. Na dit een of twee keer gedaan te hebben besprak ik mijn schetsen met iemand van

het team(meestal Vincent). Door ook deze vroege schetsen al te bespreken kon ik snel oordelen of een oplossing in de goede richting was of niet. Voor de vaste volgorde van kinderen schetste ik aanvankelijk bijvoorbeeld een aantal aangepaste “triggers”- de knoppen in de toolbar om het element aan te maken(afbeelding H). Na overleg met Vincent bleek echter dat deze oplossing voor te veel informatie in de knoppen zou zorgen en dus niet geschikt voor gebruik was.



Afbeelding H. Een vroege schets voor het oplossen van het designprobleem “vaste volgorde van kinderen bij het aanmaken van elementen”.

Om te zorgen dat iedere schets representatief zou zijn voor een echt probleem, heb ik gekeken naar de voorbeelden van elementen die in mijn onderzoek naar XML-bewerking per designprobleem worden gegeven. Voor het omgaan met vaste volgorden koos ik bijvoorbeeld om de kind-elementen van <taskbody> als uitgangspunt voor de ontwerpen te nemen. Wanneer een oplossing geschikt leek en vorm begon te krijgen, pakte ik de systeemeisen erbij. Zo kon ik tijdens het ontwerpen van een oplossing bijhouden of deze aan alle systeemeisen voldeed. Was dit niet zo, dan moesten er dingen worden aangepast. In het verder uitwerken maakte ik ook voor iedere aparte oplossing weer een beschrijving. Daarnaast stelde ik een aantal voor- en nadelen per oplossing op. Zo kon vervolgens weer met het team worden besproken of de voordelen voldoende waren en de nadelen acceptabel. De definitieve uitwerking van een ontwerp legde ik vast in een storyboard. De uitwerking voor ontwerp A bij het omgaan met vaste volgorden, inclusief beschrijving en voor- en

nadelen is te zien in het kader hieronder. Het bijbehorende storyboard is te zien in [afbeelding I](#). Dit is dus de uiteindelijke oplossing voor de schets uit [afbeelding H](#).

Ontwerp A: Automatisch juiste positie toewijzen

Ontwerp A beschrijft een aanpak waarbij FontoXML automatisch de juiste positie aan een element toewijst. In dit voorbeeld is de knop voor het aanbrengen van <context> altijd actief, zolang de cursor focus op een <task> heeft. Bij het klikken op deze knop, wordt <context> automatisch op zijn toegestane positie ingevoegd.

Voordelen

- Vergevingsgezind: de auteur hoeft de locatie niet uit zijn hoofd te weten omdat het systeem deze automatisch kiest.
- Speelt beter in op de gebruikersintentie dan wanneer de knop helemaal is uitgeschakeld.

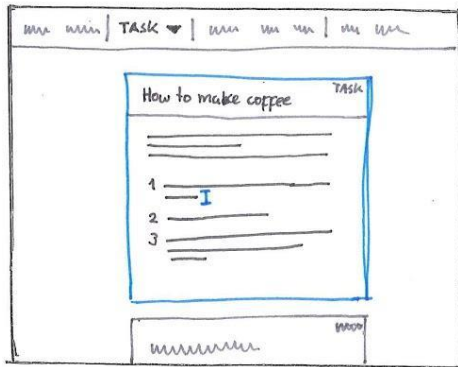
Nadelen

- Geen informatie vooraf over vaste volgorde en al helemaal niet over welke volgorde dan.
- Voert niet precies de actie uit die de gebruiker eigenlijk probeert.

Overige aandachtspunten

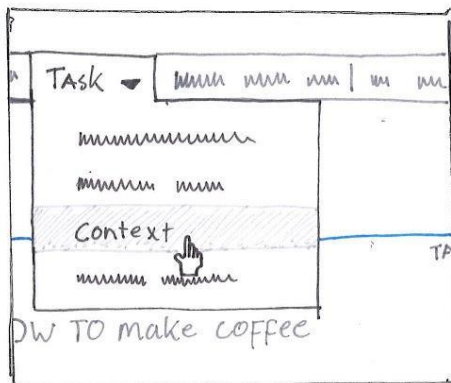
- Moeilijk te schalen: voor welke elementen doe je dit wel en voor welke niet? Dit heeft nog denkwerk nodig.
- Toon je de message iedere keer? of voor 1 of 2 keer? Of laat je de gebruiker hem "dismissen?".

Vaste element volgorde - aanmaken (1)



Situatie A: Cursor staat in <STEP>, waar <context> niet is toegestaan.

- Gebruiker klikt op **Task**



Situatie B: Ondanks het feit dat <context> niet is toegestaan op de cursorpositie, is de knop toch actief.

- Als de gebruiker op context klikt, zal deze op de meest nabije toegestane locatie worden ingevoegd.



Situatie C: context is tussen shortdesc en steps geplaatst binnen de huidige task.

Het systeem geeft een bericht ter uitleg.

*1: Feedback message.

"context" has a fixed location within a task. Therefore, it has been placed on a different place then the cursor position. undo

Afbeelding I. Het uitgewerkte storyboard voor oplossing A van het probleem "omgaan met vaste volgorde".

Zoals eerder genoemd heb ik voor de meeste designproblemen meerdere oplossingen ontworpen. Ook bij het omgaan met vaste volgorden was dit het geval. Het tweede ontwerp voegt een heel nieuw onderdeel toe aan de applicatie: een paneel met structuursuggesties. Het idee hierbij is dat gebruikers, bij vraagtekens over de juiste structuur, via de breadcrumbs een hulppaneel kunnen openen. De uitwerking van dit ontwerp is te zien in afbeelding J.

Ontwerp B: Paneel met structuursuggesties

Voor bepaalde (of alle?) elementen is de “structure guide” op te roepen in het rechter zijpaneel. Hierin staat een overzicht voor de mogelijke elementen die je binnen dat element zou kunnen plaatsen om hem verder te verrijken. Daarnaast kun je vanuit het paneel de verschillende, gesuggereerde elementen ook daadwerkelijk invoegen op de juiste positie. Ook staat er een korte uitleg over de betekenis en het gebruik van het element zelf.

Voordelen

- Veel ruimte voor uitleg over de volgorde en andere achtergrondinformatie.
- Ook toepasbaar voor andere uitleg-doeleinden, niet alleen voor de vaste volgorde.

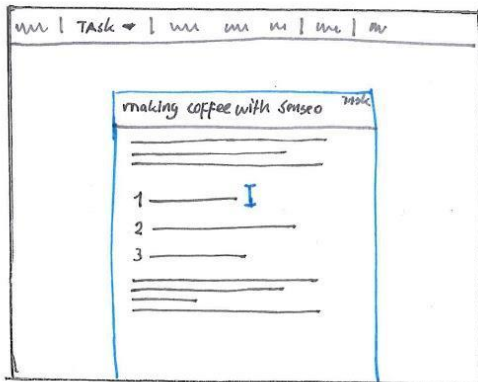
Nadelen

- Het paneel moet eerst worden geopend voordat de gebruiker iets leert over de vaste volgorde van de kind-elementen. Het loopt dus niet heel soepel mee in de aanmaak-flow.

Overige aandachtspunten

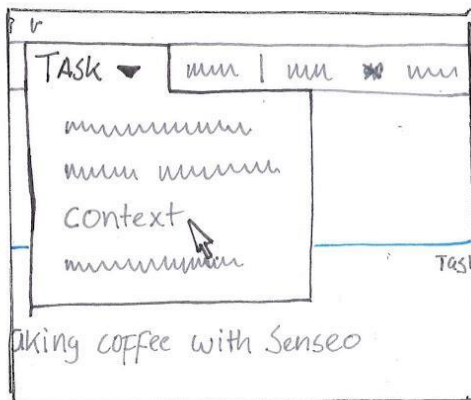
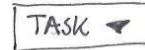
- Blijf goed nadenken over wat er wel en niet in het zijpaneel moet. Het moet niet overvol raken.
- Hoe bepaal je voor welke elementen dit kan? Je wilt niet de hele schemadocumentatie hebben toch..?

Vaste elementvolgorde - (aanmaken 2A)



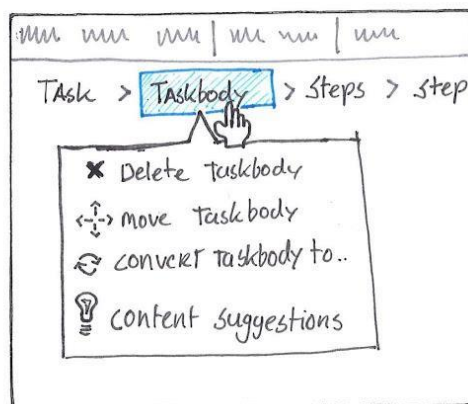
Situatie A: Cursor staat in <step>, waar <context> niet is toegestaan.

- Gebruiker klikt op

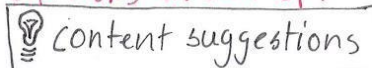


Situatie B: De knop voor het aanmaken van <context> is uitgeschakeld, vanwege een niet-toegestane positie.

- Gebruiker denkt "waarom mag dit niet?" en gaat op zoek naar de reden



Situatie C: De gebruiker klikt in de breadcrumbs op Taskbody (of task) en dan op.

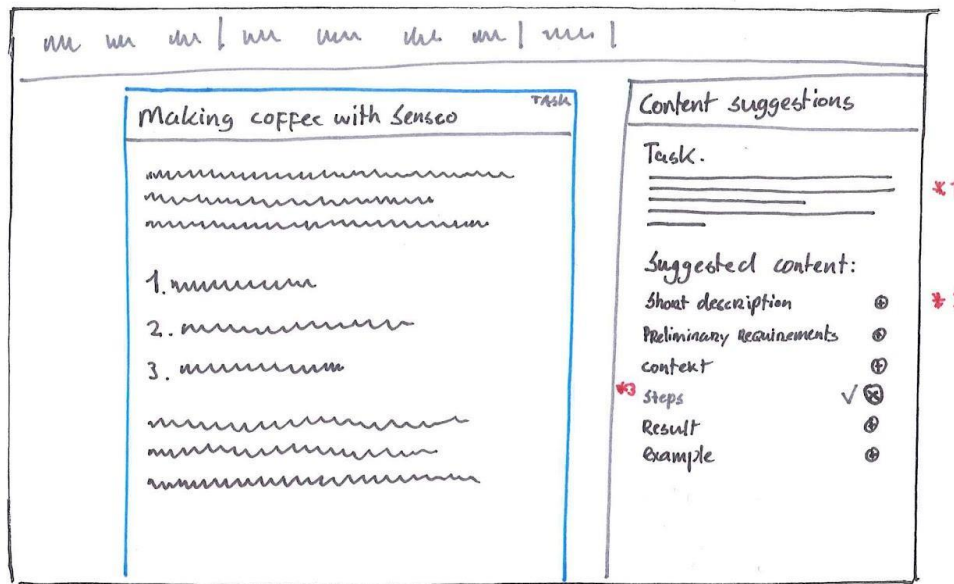


Situatie D: zie 2B



Afbeelding J-1. Deel 1 van het storyboard voor oplossing B van het probleem "omgaan met vaste volgorde".

Vaste elementvolgorde - aanmaken (2B)



Situatie D: Na het openen van de content suggestions toont het zijpaneel zich met info over <Task>.

*1: In een korte tekst kan worden uitgelegd dat we met een vaste volgorde te maken hebben.

*2: Per element biedt een (+) -knop de mogelijkheid om dit element automatisch op de goede positie in te voegen.

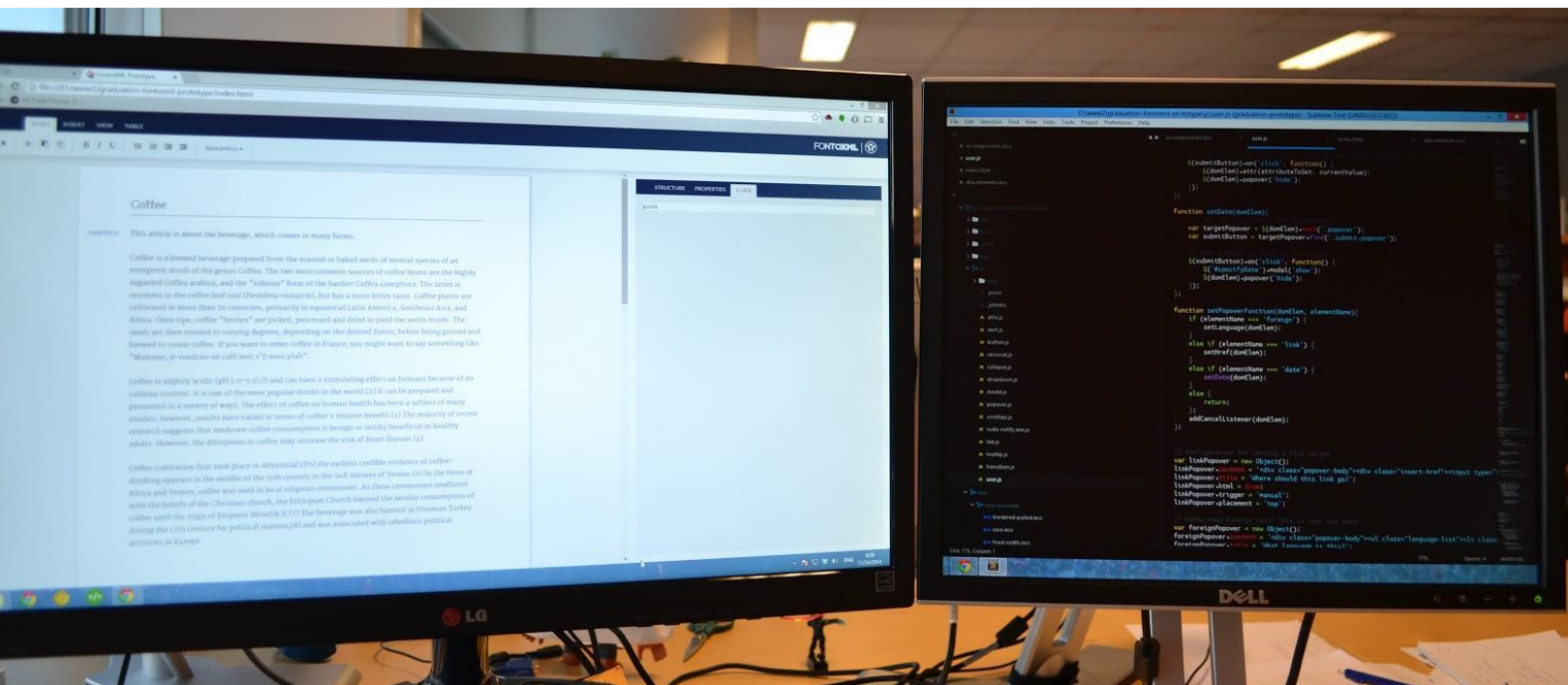
*3: Steps bestaat al, en is dus ook daar inactief

Afbeelding J-2. Deel 2 van het storyboard voor oplossing B van het probleem "omgaan met vaste volgorden".

Uiteindelijk zijn er voor ieder probleem maximaal twee ontwerpen helemaal uitgewerkt. Er is voor dit aantal gekozen omdat, met het ontwikkelen van de prototypes in het achterhoofd, werd ingeschat dat dit qua tijd het maximaal haalbare zou zijn. De uitwerking van de oplossing voor alle designproblemen is te vinden in bijlage G: Ontwerprapport.

8. Realisatiefase

De realisatiefase is bedoeld om ontwerpen om te zetten in een functionerend product. Zo kan er door het team worden beoordeeld of de ontwerpen geschikt zijn voor gebruik in de echte applicatie. Vervolgens kan er, aan de hand van het product, met gebruikers worden getest of ook zij hun taken kunnen uitvoeren met de ontworpen interfacepatronen. Het is dus van belang dat het gerealiseerde product zo dicht mogelijk bij het werkelijke eindproduct staat, zodat er maximale lering uit getrokken kan worden.



8.1. Bouwen van high-fidelity prototype

Toen de ontwerpfase eenmaal was afgerond ben ik begonnen met het ontwikkelen van een prototype. Het doel van dit prototype was om de ontworpen interfacepatronen werkend te kunnen zien, in een zo realistisch mogelijke context. Op deze manier kon worden beoordeeld en getest of de ontwerpen hun functie goed vervullen. Zonder dit prototype kon er dus niet worden getest en in de User Centered Design-methode is testen van cruciaal belang voor het bewijzen van de gebruiksvriendelijkheid van een ontwerp. De laatste (werkende) versie van het prototype is te bekijken en te gebruiken op www.yourboss.nl/fontoxmlprototype.

8.1.1. Aanpak bepalen

Voordat de ontwikkeling van het prototype kon beginnen moest er ook hier eerst een aanpak gekozen worden. Er zijn veel verschillende soorten prototypes en veel verschillende technieken die voor de ontwikkeling daarvan gebruikt kunnen worden. Zo is er allereerst onderscheid te maken in de fidelity(nauwkeurigheid) van een prototype. Al bij de planning van dit project had ik vastgesteld een high fidelity prototype te willen ontwikkelen. Deze keuze heb ik gemaakt omdat dit het meest representatieve resultaat oplevert. Door een prototype nauwkeuriger en meer waarheidsgetrouw te maken zullen de resultaten van eventuele tests uitgebreider en gedetailleerder zijn. Op deze manier kan er dus het meeste uit een test gehaald worden.

Ook voor een high fidelity prototype zijn echter nog meerdere technieken mogelijk. In mijn geval waren er drie goede opties: een feature branch maken van de echte applicatie FontoXML, een HTML-prototype bouwen, of een prototype maken met een prototyping-tool. Voor elke techniek geef ik kort uitleg wat deze inhoudt en waarom ik deze wel of niet heb toegepast.

1. *Feature branch*

Mijn aanvankelijke idee voor het prototype was om een feature branch te maken op een bestaande versie van FontoXML. De feature branch is een begrip uit het populaire versiebeheersysteem “Git”. Simpel gezegd zou dit inhouden dat ik online een plek zou maken waar ik de bestaande applicatie van FontoXML vrijuit zou kunnen aanpassen. Op deze manier zou de rest van het team of de klanten daar geen last van hebben. Een groot voordeel hiervan zou zijn geweest dat ik in de applicatie mijn ontwerpen kon testen. Hierbij zou ik dus beschikken over alle functies die FontoXML op dat moment bevatte, zonder dat ik daarvoor iets hoefde te doen. Dit was veruit de meest representatieve manier van prototypen geweest. Een enorm nadeel van deze aanpak was echter dat ik de Javascript van FontoXML hiervoor aan zou moeten gaan passen. Deze code was zeer diep geworteld in een ingewikkelde structuur. Het uitvogelen hiervan had zowel mij als het team enorm veel tijd

gekost. Om deze reden ben ik afgeweken van mijn planning en heb ik ervoor gekozen geen feature branch van de bestaande applicatie te maken.

2. *HTML-prototype*

De tweede kandidaat-techniek was het bouwen van een HTML-prototype. Hiervoor zou ik in HTML een applicatie moeten bouwen die sterk lijkt op FontoXML, zodat ik daarin mijn ontwerpen kon testen. Deze aanpak kost aanvankelijk behoorlijk wat meerwerk, omdat de gehele layout gereproduceerd zou moeten worden. Vervolgens is er echter wel veel meer vrijheid voor het toevoegen van nieuwe functies, doordat er geen basiscode is die zal moeten worden aangepast. Dit was voor mij een reden die sterk genoeg was om voor deze techniek te kiezen.

3. *Prototyping-tool*

De laatste mogelijke techniek was het gebruiken van een prototyping tool. Voorbeelden van zulk soort tools zijn Marvel(marvelapp.com), Invision (invisionapp.com) en Justinmind(justinmind.com). Met zo'n tool kunnen schetsen of mockups snel en gemakkelijk worden omgezet naar interactieve, klikbare prototypes. Een groot voordeel van dit soort tools is dat je ontwerpen rechtstreeks om kunt zetten naar prototypes: je hoeft dus geen code te kunnen schrijven. In mijn geval was dit echter geen erg sterk voordeel: ik had geen mockups, en ik kan wel code schrijven. Met die code kon ik een stuk gedetailleerder interactie toevoegen aan mijn prototype en hoefde ik niet alsnog mockups te gaan maken.

Om die reden heb ik niet gekozen voor het gebruiken van een tool.

Uiteindelijk koos ik er voor om een HTML-prototype te bouwen. Deze was minder waarheidsgetrouw dan een feature branch op de echte applicatie, maar bood mij wel veel meer vrijheid en nog steeds een goed beeld van de echte situatie. Deze vrijheid zou mij veel kostbare tijd besparen en dus meer tijd overlaten voor het coderen van gedetailleerde interacties.

8.1.2. Bouwen van het prototype

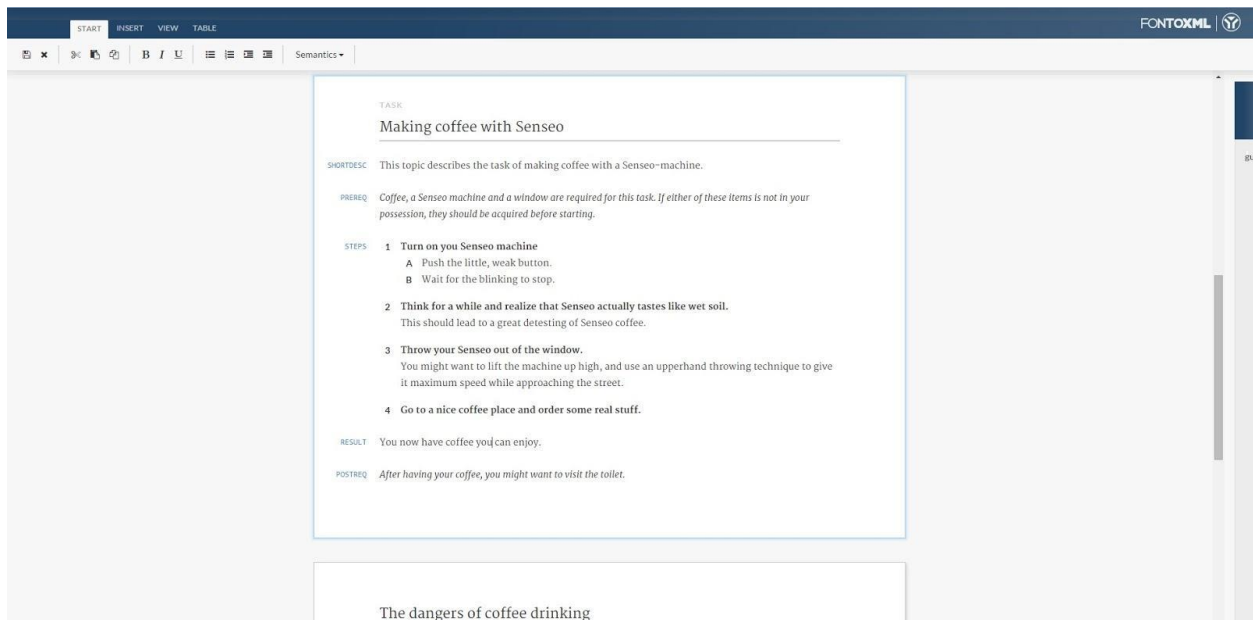
Toen mijn keuze eenmaal was gemaakt, ben ik begonnen met het opzetten van de basis voor mijn prototype. Hierbij heb ik allereerst een Git-repository opgezet zodat ik op verschillende plekken aan mijn prototype zou kunnen werken zonder gegevens te verliezen. Ook levert dit versiebeheersysteem grote voordelen op indien er fouten worden gemaakt: deze zijn gemakkelijk weer terug te draaien. De keuze voor Git was hierbij snel gemaakt: ik was bekend met de werking en het heeft voor mij altijd naar behoren gefunctioneerd. Hierdoor had ik geen behoefte om naar een alternatief te zoeken voor dit project.

Toen het versiebeheer eenmaal stond ben ik op zoek gegaan naar een snelle manier om de layout van FontoXML na te bouwen. Hiervoor dacht ik al snel aan Twitter Bootstrap. Dit front-end framework biedt een snelle manier om een webapplicatie op te zetten, met een aantal vooraf

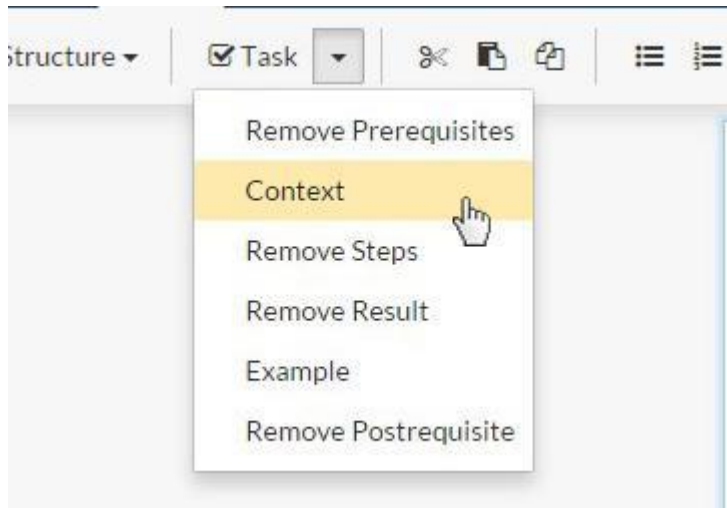
gedefinieerde interfacecomponenten die vaak voorkomen in websites en webapplicaties, zoals modal windows, popovers en tooltips. Een ander bijkomend voordeel is dat jQuery al zit ingebakken in Bootstrap. Ik was al van plan om dit Javascript framework te gaan gebruiken vanwege de toegankelijkheid en de vele functies voor DOM-manipulatie. In het bewerken van documenten komt namelijk veel DOM-manipulatie voor. Het feit dat Bootstrap standaard jQuery bevat, was dus alleen maar een pre.

Met deze frameworks was mijn setup voor het project compleet. Allereerst ben ik toen de interface van FontoXML gaan nabouwen. Dit deed ik door mijn prototype en de echte applicatie naast elkaar weer te geven op mijn twee schermen. Via de developertools van mijn browser achterhaalde ik de maten die in FontoXML gehanteerd werden en kon ik deze snel overnemen. Ik heb ervoor gekozen om alleen de onderdelen die relevant zijn voor mijn prototypes en de onderdelen die heel snel geproduceerd konden worden, na te bouwen. Dit heb ik gedaan omdat het nabouwen van andere functies geen (voor mij) relevante testresultaten zou opleveren. Ondanks het feit dat het een meer waarheidsgetrouwe omgeving zou opleveren, zou het mij dus niets over de gebruiksvriendelijkheid van mijn ontworpen patronen leren.

Voor het uitwerken van alle oplossingen heb ik het ontwerprapport als uitgangspunt genomen. Omdat hierin zowel de storyboards als de onderbouwing was opgenomen, was dit een goed hulpmiddel bij het ontwikkelen van het prototype. Zo ben ik voor “vaste volgorde van kinderen” begonnen met het nabootsen van de ingangssituatie. In het HTML-document heb ik een <task> gemaakt met daarin een aantal van de onderdelen die daarbij horen ([afbeelding K](#)). Het onderdeel <context> heb ik bewust weggelaten, zodat het plaatsen daarvan kon worden getest.



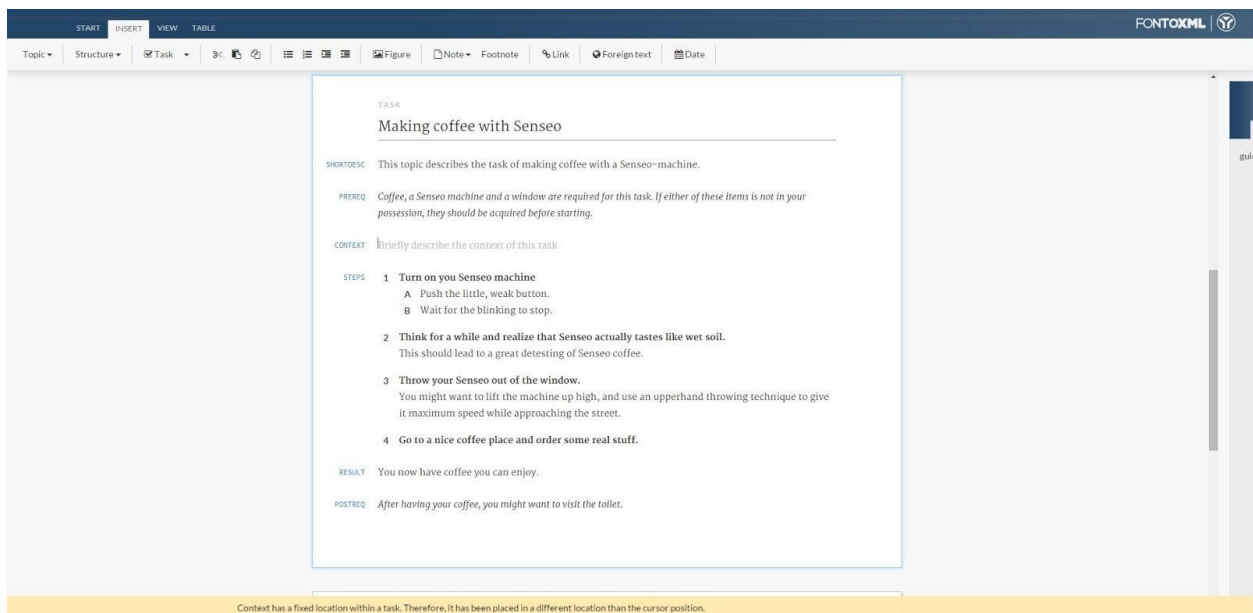
Afbeelding K. Het prototype met daarin een bewerkbare “task”. De cursor staat in het onderdeel “result”.



Als je met de cursor in de <task> gaat staan, wordt de dropdown voor het invoegen van de bijbehorende onderdelen actief. Ook de knop voor het invoegen van <context> is geactiveerd (afbeelding h), ondanks het feit dat je daarvoor niet op een toegestane positie staat.

Afbeelding h. De knop voor het aanbrengen van "context" is actief, ook al mag deze niet bestaan op de huidige cursorpositie.

Vervolgens heb ik in de jQuery vastgelegd in welke volgorde de verschillende onderdelen van <task> dienen te staan. Bij een klik op de knop kijkt het systeem of er al andere onderdelen bestaan en plaatst de <context> op de juiste positie. Hier verschijnt een placeholder tekst waarin bondig wordt geïnstrueerd wat er in dit veld gedaan moet worden. De cursor wordt verplaatst naar het nieuwe element zodat hiervoor direct inhoud geschreven kan worden. Ook verschijnt er onderin beeld een melding waarin wordt uitgelegd waarom dit element op een afwijkende locatie is geplaatst in plaats van op de cursorlocatie (afbeelding L). Tijdens het bouwen van het prototype heb ik weinig aandacht besteed aan het schrijven van schone code. Zolang de interface werkte zoals ontworpen was het goed: de interacties werden hier immers geprototyped, niet de code.

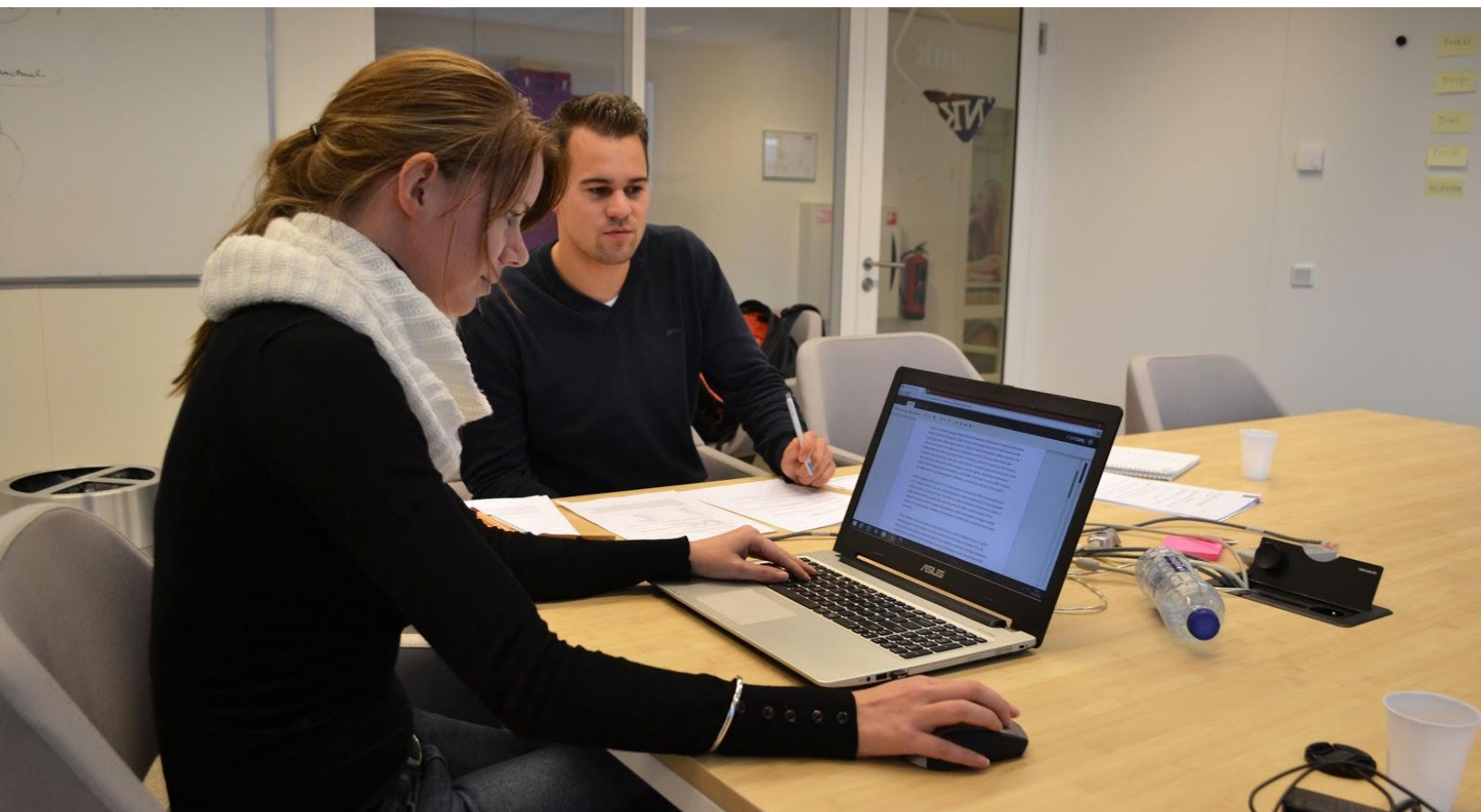


Afbeelding L. De "context" is op de juiste positie verschenen. Een systeembericht (onderin) legt uit wat er gebeurd is.

Als alle beschreven en geschetste interactie uit de ontwerpen in het prototype was geïmplementeerd, rondde ik de functie af en uploadde ik deze naar mijn website. De meest recente werkende versie van het prototype is dan ook te vinden op www.yourboss.nl/fontoxmlprototype. Gezien het feit dat het hier gaat om een interactieontwerp, is dit de beste plaats om het product te bekijken. Mocht deze om een bepaalde reden niet beschikbaar zijn, dan is er voor elk van de uitgewerkte designproblemen ook nog een aantal screenshots van het prototype te zien in bijlage H: Screenshots prototype.

9. Testfase

De testfase draaide geheel om het controleren van de kwaliteit van de gemaakte ontwerpen. Dit gebeurde geheel aan de hand van het prototype dat was ontwikkeld in de realisatiefase. In een zo realistisch mogelijke situatie moest gekeken of gebruikers hun taken kunnen vervullen en hun doelen kunnen bereiken met behulp van de ontwikkelde oplossingen. Pas als deze tests waren geslaagd, kon daadwerkelijk worden gezegd dat er “goede, gebruiksvriendelijke” ontwerpen waren gemaakt. De resultaten van deze tests konden verder ook leiden tot een aantal mogelijke verbeterpunten in de ontworpen oplossingen. Als laatste kon er, gebaseerd op deze resultaten, een advies worden opgesteld aan de klant over het toepassen van de verschillende oplossingen binnen FontoXML.



9.1. Voorbereiden van de usabilitytest

Ter voorbereiding van de usabilitytest die ik zou gaan houden heb ik een aantal activiteiten uitgevoerd. De belangrijkste van deze activiteiten staan in deze paragraaf beschreven.

9.1.1. Doelen vastleggen

Om te beginnen heb ik voor mijn test een aantal doelen geformuleerd. In deze doelen beschrijf ik wat ik met mijn test hoop te bereiken. Ik kwam hierbij tot de volgende drie doelen:

1. Controleren of gebruikers met mijn ontwerpen de bijbehorende taken succesvol kunnen uitvoeren.
2. Controleren of gebruikers te allen tijde begrijpen wat er gebeurt tijdens het uitvoeren van de testtaken.
3. Ontdekken van verbeterpunten in mijn ontwerpen.

9.1.2. Testpersonen werven

Na de doelen te hebben vastgesteld, ben ik gaan nadenken over de testpersonen die ik nodig zou hebben. Uit eerdere ervaring had ik geleerd dat het werven van testpersonen doorgaans veel tijd kost, dus wilde ik er dit keer snel bij zijn.

Voordat ik ging werven, stelde ik vast aan welke eisen deze testpersonen zouden moeten voldoen om representatief te zijn. Ik had al bedacht dat ik, voorafgaand aan de tests een soort “proeftests” wilde gaan doen. Met deze proeftests kon ik controleren of ik er nog verbeterpunten in de test zelf zouden zitten. Er zouden dus eigenlijk twee varianten van de test komen: de “proeftest” en de “echte” test. Voor deze varianten wilde ik ook verschillende gebruikers hebben. Omdat de proeftest niet zozeer was gericht op het verzamelen van waardevolle testresultaten, was het hierbij niet nodig om “echte” gebruikers als testpersoon te werven. Wel stelde ik de volgende eisen op voor deze testpersonen:

- Benodigde testpersonen: 2
- Leeftijd: 25 - 65 jaar
- Geen kennis van XML
- Niet eerder met FontoXML gewerkt
- Gewend aan het schrijven van documenten in Microsoft Word

Ik zocht dus mensen die wel enigszins leken op gebruikers, maar gemakkelijker te vinden en te werven waren. Ik heb hierbij gekozen voor collega's van Liones die nog niet eerder met FontoXML hadden gewerkt. Ik wist namelijk zeker dat zij dezelfde vakkennis zouden hebben, waarvoor ik ook gemakkelijk testcontent kon creëren.

Voor de “echte” tests ben ik wel op zoek gegaan naar representatieve gebruikers. Het is altijd beter om te testen met dit soort gebruikers (Spool, 2005) omdat hun eigenschappen en gebruiken invloed hebben op de manier waarop ze jouw applicatie gebruiken. Om te kijken of dit mogelijk zou zijn heb ik via collega Bert contact opgenomen met klant ThiemeMeulenhoff. Gebaseerd op het boek “Don’t make me think” (Krug, 2011) heb ik bij hem aangegeven met minimaal 4 mensen te willen testen. Verder had ik voor deze testpersonen de volgende eisen opgesteld:

- Leeftijd: 25 - 65 jaar
- Geen kennis van XML
- Niet eerder met FontoXML gewerkt
- Beroepsmatig bezig met het schrijven van lescontent
- Werkzaam bij ThiemeMeulenhoff
- Gewend aan het schrijven van documenten in Microsoft Word

Het is dus zo dat ik mij bij het testen alleen heb gericht op vakexperts (de primaire gebruikersgroep) en niet op invoerders (de secundaire doelgroep). Dit heb ik gedaan omdat ik geleerd had dat de invoerders een gebruikersgroep zijn die waarschijnlijk alleen bij ThiemeMeulenhoff bestaat. Bij alle klanten zullen vakexperts met FontoXML gaan werken, omdat zij de daadwerkelijke auteurs zijn. Om de resultaten van mijn test zo breed mogelijk toepasbaar te houden, heb ik de test dus beperkt tot de vakexperts.

Nadat deze Bert deze vraag had gesteld bij ThiemeMeulenhoff, gaven zij aan inderdaad vier testpersonen tot mijn beschikking te willen stellen. Toen deze afspraken waren gemaakt, ging ik verder met het voorbereiden van mijn tests.

9.1.3. Methode kiezen

Bij het uitvoeren van een test is het gebruikelijk om een bepaalde methode te gebruiken. Dit biedt houvast in de stappen die je moet uitvoeren. Gebaseerd op de doelen en onderzoeksvragen van mijn test koos ik voor de methode “Comparison Test”. Deze methode is geschikt wanneer er verschillende ontwerpen moeten worden getest die in dezelfde behoefte moeten voorzien (Rubin, 2008). Door beide ontwerpen op een kwalitatieve manier te testen zouden er voor- en nadelen van ieder ontwerp gevonden kunnen worden. Aan de hand van deze voor- en nadelen kon ik dan een advies opstellen voor de opdrachtgever.

Als ondersteuning bij deze methode heb ik gekozen om de techniek “Thinking Aloud” te gebruiken. Hierbij moeten testpersonen een aantal taken uitvoeren en daarbij hardop hun gedachten uitspreken. Deze veelgebruikte techniek biedt inzicht in het denkproces van gebruikers, wat veel waardevolle informatie kan opleveren voor het verhelpen van eventuele usabilityproblemen of knelpunten (Nielsen, 2012).

9.1.4. Benodigdheden in kaart brengen

Om te voorkomen dat ik bij aanvang van de test voor verrassingen zou komen te staan, heb ik ook de benodigdheden voor de test in kaart gebracht. Het belangrijkste hierbij was om te beschrijven wat de *testpersonen* nodig zouden hebben om de test goed uit te kunnen voeren. Hiervoor kwam ik tot twee punten:

1. Handout met vereiste kennis

Er is een bepaalde basiskennis die nodig is om met FontoXML gestructureerde content te kunnen bewerken. Uit eerdere gebruikersonderzoeken is gebleken dat de meeste gebruikers niet over deze kennis beschikken. Om de tests succesvol uit te kunnen voeren, zal deze basiskennis dus vooraf aan de gebruiker moeten worden aangeboden. Het is van belang dat hier zo min mogelijk informatie overgedragen hoeft te worden. De gebruiker moet niet het gevoel krijgen dat hij deze kennis “zou moeten weten”. Om die reden moet deze kennis op een handout worden aangeboden, die eventueel tijdens de test kan worden geraadpleegd. De handout is vinden in Bijlage A: Handout gebruikerstest. Deze handout wordt na de mondelinge introductie aan de testpersoon overhandigd.

2. Begrijpelijke testcontent

In de werkelijkheid zullen testpersonen FontoXML gaan gebruiken voor het schrijven en bewerken van documenten die met hun vakgebied te maken hebben. De content en de betekenis daarvan is in die situatie dus vertrouwd en begrijpelijk. Om deze situatie na te bootsen dient er, bij het uitvoeren van de taken, testcontent te worden gebruikt die te maken heeft met het vakgebied van de gebruiker. Deze testcontent moet vooraf worden verzameld en klaargezet in de applicatie.

Ik schreef daarna een handout met daarop een introductie en de uitleg van twee begrippen waarvan ik verwachtte dat de testpersonen deze wellicht niet zouden kennen. Vervolgens heb ik voor beide gebruikersgroepen een aantal stukken testcontent samengesteld. Voor mijn collega's heb ik die zelf geschreven en voor de testpersonen van ThiemeMeulenhoff heb ik door hen gepubliceerd lesmateriaal gebruikt.

9.1.5. Taken opstellen

Tot slot heb ik een lijst met testtaken opgesteld. Voor ieder designprobleem heb ik een testtaak bedacht waarmee ik kon zien of mijn ontwerp de juiste oplossing bood voor dit probleem. Per testtaak heb ik een aantal onderdelen beschreven:

1. *Gebruikersinstructie*: De opdracht die de gebruiker krijgt
2. *Testelementen*: De elementen waarmee deze testtaak uitgevoerd zou kunnen worden.
3. *Ideale flow*: De snelste/beste manier om de taak uit te voeren.

4. *Aandachtspunten*: De factoren die bepalen of het ontwerp geschikt is.

Ter illustratie is in de kadertekst hieronder de testtaak opgenomen voor het designprobleem *“Invoegen van een element met vaste volgorde”*.

Taak 3: Invoegen van een element met vaste volgorde

Gebruikersinstructie

“Ergens in het document vind je een taak. Schrijf voor deze taak een mogelijk resultaat.”

Testelement(en)

<result>, <context>, <prereq>, <postreq>, <example>

Ideale flow

1. Focus task
2. Insert-tab
3. Task-dropdown
4. Klik “Result”
5. Begin te typen
6. *Taak voltooid*

Aandachtspunten

Aandachtspunt	Commentaar
Vindt de gebruiker de Task?	
Waar gaat hij met de cursor staan?	
Ziet hij waar het element is ingevoegd?	
Ziet hij de feedback message?	
Kan de gebruiker benoemen wat er gebeurd is?	
Overig	

9.1.6. Uitvoeren van proeftest

Als voorbereiding op de test met de representatieve deelnemers, heb ik eerst een aantal proeftests uitgevoerd. Hiermee wilde ik ontdekken welke verbeterpunten er nog in de test zelf te vinden

waren. Op deze manier zou ik die punten nog kunnen verbeteren voordat ik de “echte” test ging uitvoeren. Voor deze tests heb ik een aantal collega’s gebruikt als deelnemers. Ik heb hen de test exact zo laten uitvoeren als ik dat met de echte deelnemers zou doen. Allereerst gaf ik een korte introductie over gestructureerd schrijven. Hierbij legde ik onder andere het concept “topic” (een bepaalde eenheid van informatie uit het DITA-schema) uit, omdat ik dacht dat dit nodig zou zijn voor het uit kunnen voeren van de taken. Vervolgens vroeg ik hen de taken uit te voeren. Tijdens de test lette ik allereerst op of zij de vragen begrepen. Of zij de testtaak vervolgens goed uitvoerden was ook interessant, maar niet het hoofddoel voor deze test.

Van een aantal vragen ontdekte ik dat deze te suggestief of te onduidelijk waren geformuleerd. Zo bleek bijvoorbeeld dat taak 2 (*“Beschrijf onder in het eerste topic, in een Hazard Statement de gevaren van koffie drinken”*) verwarrend was voor de deelnemers. Toen ik dit ontdekte heb ik twee dingen veranderd. Ten eerste heb ik de uitleg- en het gebruik van de term “topic” geschrapt uit mijn test. Het was voor mijn taken immers niet van belang of mensen dit wel of niet begrepen. Daarom heb ik de locatie waar het hazard statement moest komen simpelweg aangewezen. De nieuwe formulering van de testtaak werd als volgt: *“Het eerste stuk / deel gaat over koffie. Zorg dat er onder in dat stuk een “gevaarbeschrijving” komt te staan, waarin de gevaren van koffie beschreven worden”*.

Verder week ik bij het stellen van de vragen soms wat af van de formulering zoals ik deze had opgeschreven. Ook dit bleek een slecht idee te zijn voor bij de test: als je eerst zorgvuldig je taken hebt geformuleerd zodat ze precies beschrijven wat je bedoelt, moet je je daar vervolgens ook aan houden. Het laatste punt wat ik uit de proeftest leerde was dat je expliciet aan deelnemers moet vragen om aan te geven wanneer zij klaar denken te zijn met een taak. Zo voorkom je dat je onnodig dezelfde vraag blijft herhalen. Ook krijg je een duidelijker beeld van de manier waarop een deelnemer naar een taak kijkt.

9.1.7. Testplan

De hierboven genoemde activiteiten vormden samen de inhoud van het testplan dat ik heb geschreven. De verschillende hoofdstukken hiervan heb ik gebaseerd op het boek “Handbook of usability testing” (Rubin, 2008), waarin uitgebreid wordt beschreven hoe je een goede usabilitytest kunt uitvoeren. Dit plan heb ik opgesteld om een aantal redenen. Ten eerste moest ik duidelijk krijgen hoe ik precies mijn testdoelen wilde bereiken. Het testplan zou dus dienen als “handboek” bij het uitvoeren van de tests. Op deze manier zou ik de test telkens op dezelfde manier kunnen uitvoeren. Daarnaast kon ik met behulp van het testplan mijn aanpak bespreken met Vincent. Als laatste diende het testplan om inzichtelijk te maken welke benodigdheden er waren voor het kunnen uitvoeren van de tests. Het gehele testplan is te vinden in [bijlage I](#).

9.2. Testen van ontworpen interfacepatronen

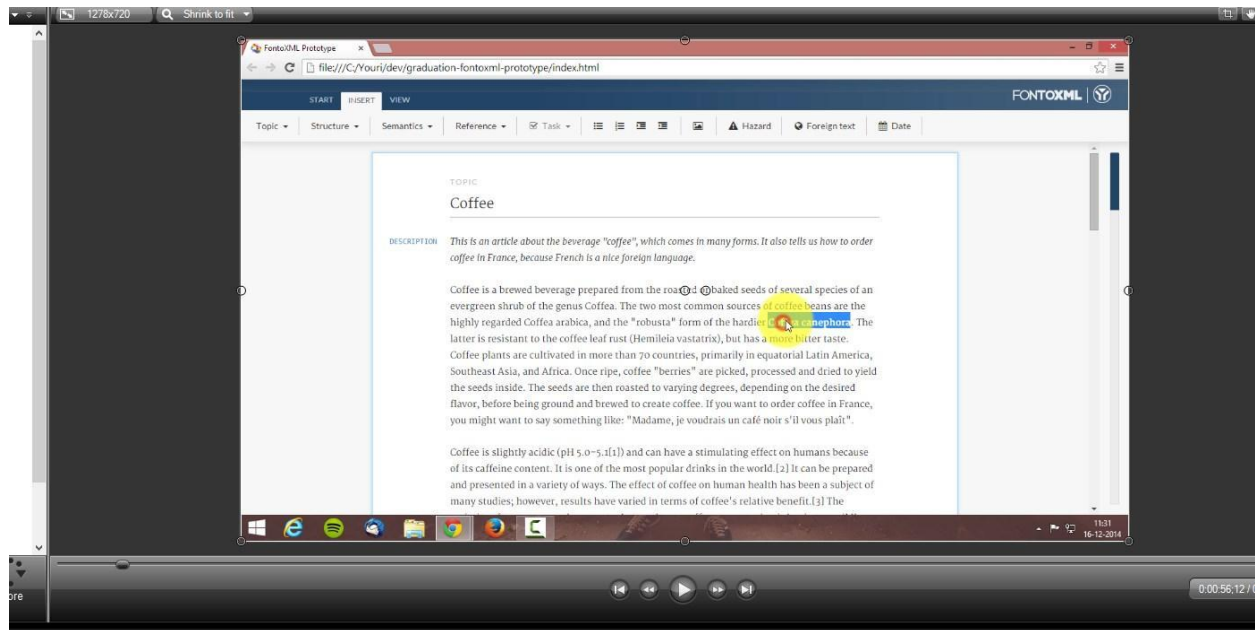
Op 16 december heb ik een viertal gebruiksvriendelijkheidstests uitgevoerd. Dit heb ik gedaan bij ThiemeMeulenhoff, een van de klanten van FontoXML waarmee er tijdens mijn project een implementatietraject werd doorlopen. Deze test had twee doelen:

1. Controleren welke van mijn ontworpen oplossingen het meest geschikt was voor implementatie in FontoXML.
2. Ontdekken van mogelijke verbeterpunten in de ontworpen oplossingen.

Om dit te achterhalen heb ik de tests uitgevoerd zoals gepland in het eerder opgestelde testplan. Een afwijking op dat testplan was dat ik de test heb gehouden samen met collega-afstudeerder Mike Zuidgeest. Ook hij moest voor zijn afstudeerproject testen bij ThiemeMeulenhoff en na enig overleg hebben wij besloten onze tests te combineren. Dit bood voor ons een aantal voordelen. Ten eerste scheelde het tijd voor de klant: door één keer te testen in plaats van twee keer, kon er een efficiënt schema gemaakt worden en hoefden de deelnemers niet tweemaal op te komen dagen. Een ander voordeel was dat Mike en ik elkaar konden gebruiken als extra observant bij de test. Op deze manier konden er nog meer punten gevonden worden en ging er geen informatie verloren als er iemand even een notitie aan het maken was. Vier ogen zien meer dan twee.

Wij hebben de tests gecombineerd, maar wel bewust twee gescheiden delen aangehouden en dit ook als dusdanig gecommuniceerd naar de klant. Wij vertelden hen dat er eerst een test zou komen over de visualisatie van semantiek (Mike's deel) en vervolgens over het aanmaken van elementen (mijn deel). Voor iedere deelnemer hebben wij hetzelfde schema aangehouden. Eerst gaf ik een algemene introductie waarin het doel van de test werd uitgelegd en de deelnemer op zijn gemak werd gesteld. Daarna gaf ik aan dat de test uit twee delen zou bestaan en begonnen wij met Mike zijn deel. Hierbij nam hij de leiding en diende ik als tweede observant.

Halverwege de test begon mijn deel, waarbij ik de leiding nam en Mike observeerde. Hier legde ik kort uit hoe de test werkte en waar de deelnemers op moesten letten tijdens het uitvoeren van de testtaken. Vervolgens gaf ik hen een aantal testtaken waarbij ik hen telkens vroeg om aan te geven wanneer zij klaar dachten te zijn. Bij vragen heb ik bewust geprobeerd om de testresultaten niet te beïnvloeden. Ik heb dan ook niet op alle vragen die deelnemers hadden antwoord gegeven. Tijdens de tests heb ik een screenrecorder gebruikt. Hiermee werd zowel alle activiteiten op het scherm als de audio opgenomen. Met het programma dat ik hiervoor heb gebruikt (Camtasia) werden ook de muisbewegingen en clicks geregistreerd (afbeelding M). Hierdoor kon ik achteraf goed analyseren waar gebruikers allemaal geprobeerd hadden te klikken.



Afbeelding M. De gele cirkel geeft in Camtasia aan waar de cursor is. Het rode kringeltje verschijnt wanneer de gebruiker klikt met de linker muisknop. Bij de rechtermuisknop verschijnt er een blur.

Ook had ik bij iedere testtaak een aantal aandachtspunten opgesteld (zie 9.1.5. voor een voorbeeld). Voor elk van deze aandachtspunten schreef ik de uitkomst op. De punten die wij als observanten tijdens de test hebben gevonden, zijn achteraf aan mij overhandigd door Mike. Daarnaast had ik de video-opnames van Camtasia beschikbaar. Met behulp van deze opnames kon ik rustig het verloop van alle tests nogmaals terugkijken en -luisteren. Op basis van mijn notities en de opnames kon ik vervolgens een testrapport opstellen, waarin de resultaten werden verwerkt en geanalyseerd. Naast opvallende acties of gedachten van gebruikers, heb ik ook zwakke punten van de test zelf genoteerd. Ook deze zijn opgenomen in het testrapport om te zorgen dat deze punten in de toekomst voorkomen zouden kunnen worden.

9.3. Uitwerken en analyseren van testresultaten

Met het uitvoeren van de test had ik een hoop resultaten verzameld. Resultaten die op zichzelf nogal onoverzichtelijk en nietszeggend waren. Daarom moesten deze resultaten worden uitgewerkt en geanalyseerd. Over deze gegevens konden dan conclusies worden getrokken, die weer als basis voor een advies konden worden gebruikt.

Om te beginnen heb ik hiervoor de opnames van de verschillende tests uitgeschreven. Ik heb alle tests teruggekeken en -geluisterd en hierbij alle handelingen, opmerkingen en opvallende punten genoteerd. Veel werk, maar het is beter dan telkens opnieuw de opnames bekijken.

Slagingspercentage

Bij het analyseren ben ik begonnen met het maken van een tabel waarin per gebruiker voor iedere testtaak te zien is of deze was voltooid. Dit kon worden gebruikt als eerste grove indicatie voor de gebruiksvriendelijkheid van ieder ontwerp. Natuurlijk geeft dit geen volledig beeld: factoren als tijd, begrip van de situatie en tevredenheid zijn hierin niet meegenomen. Hier ging het enkel om succes: heeft de deelnemer de desbetreffende taak uit kunnen voeren met een correct resultaat tot gevolg? Dit is dus niet per se een indicator voor gebruiksvriendelijkheid, maar eerder voor bruikbaarheid.

Slagen van testtaken				
	Tineke	Dave	Lillian	Rindert
Taak 2 - modal	Ja	Ja	Ja	Ja
Taak 2 - template	Ja	Ja	Ja	Ja
Taak 3 - panel	Nee	Nee	Nee	Ja
Taak 3 - auto	Ja	Ja	Ja	Ja
Taak 4 - convert	Ja	Ja	Ja	Ja
Taak 4 - radio	Ja	Nee	Ja	Ja
Taak 5 - disable	kan niet slagen			
Taak 5 - remove	kan niet slagen			
Taak 6 - popover	Ja	Ja	Ja	Ja
Taak 7 - modal	Ja	Ja	Ja	Ja
Taak 7 - in trigger	Ja	Ja	Ja	Nee

Begrip van de situatie

Een van de richtlijnen die ik had opgesteld voor mijn ontwerpen was de volgende: “*Verklaar onverwachte situaties en situaties die XML-kennis vereisen*”. Wanneer er sprake was van restricties of regels in de content, moet dit zo duidelijk mogelijk aan de gebruiker worden uitgelegd. Mijn idee was hier dat, door twijfel en onbegrip weg te nemen, frustratie voorkomen zou worden. Hierom was begrip van de situatie een van de andere belangrijke meetpunten voor de gebruiksvriendelijkheid van mijn ontwerpen. In de aandachtspunten had ik hier vaak al rekening mee gehouden, maar ook hiervan wilde ik graag een overzicht hebben. Om die reden heb ik ook voor deze factor een overzichtstabel gemaakt. In de tabel is voor ieder ontwerp te zien of de deelnemer achteraf kon benoemen met wat voor situatie hij te maken had.

Begrip van de situatie				
	Tineke	Dave	Lillian	Rindert
Taak 2 - modal	Ja	Nee	Nee	Ja
Taak 2 - template	Nee	Nee	<i>Niet te beoordelen</i>	<i>Niet te beoordelen</i>
Taak 3 - panel	Nee	Ja	Nee	Nee
Taak 3 - auto	Ja, maar twijfelt	Ja	Ja, maar twijfelt	Ja, maar twijfelt
Taak 4 - convert	Ja	<i>Niet te beoordelen</i>	Ja	Nee
Taak 4 - radio	Ja	Nee	Ja, maar twijfelt	Ja
Taak 5 - disable	Ja	Ja, maar twijfelt	Ja	Ja
Taak 5 - remove	Ja, maar twijfelt	Ja, maar twijfelt	Ja, maar twijfelt	Nee
Taak 6 - popover	Ja	Ja	Ja	Ja
Taak 7 - modal	Ja	Ja	Ja	Ja
Taak 7 - in trigger	Nee	Ja	<i>Niet te beoordelen</i>	Nee

Opvallende observaties

Voor sommige testtaken waren er, buiten de vooraf opgestelde aandachtspunten, nog andere opvallende observaties die ik vast wilde leggen. Hiervoor heb ik een apart hoofdstuk opgesteld, waarin deze observaties worden beschreven. Zo viel het mijn bijvoorbeeld op dat er tijdens het zoeken naar knoppen in de toolbar zelden labels van dropdowns gelezen werden. In plaats van dat mensen van de “meest logische” naar de “minst logische” dropdown navigeren, bewegen zij vaak simpelweg van links naar rechts of andersom. Dit wijkt erg af van de verwachtingen die ik had(omdat mijn eigen zoekgedrag anders is) en daarom vond ik dit een waardevol testresultaat.

Conclusies

Na het analyseren van de testresultaten heb ik per designprobleem een aantal conclusies getrokken. Hieruit blijkt welke van de twee ontwerpen naar mijn idee het meest gebruiksvriendelijk is. Daarnaast geef ik aan of het beste ontwerp al voldoende gebruiksvriendelijk is, of dat het nog verbeteringen nodig heeft die voor implementatie moeten worden opgelost. Elk van deze conclusies heb ik onderbouwd met een aantal testresultaten.

Verbetervoorstellen

Voor een aantal van de ontwerpen kwamen er tijdens de test verbeterpunten naar boven. Bij de conclusies zijn deze verbeterpunten aangegeven. In sommige gevallen is daarvoor al direct een verbetervoorstel gedaan. Dit heb ik alleen gedaan als de oorzaak van het verbeterpunt heel duidelijk uit de test naar voren is gekomen en de oplossing daardoor voor de hand lag, zoals bij het aanmaken van elementen met een vaste positie. Hier deed ik het volgende voorstel:

Een van de opvallende punten voor het ontwerp van elementen op een vaste positie was dat systeemberichten bij het aanmaken van elementen niet worden gelezen. Dit komt omdat gebruikers volledig focussen op het document: zij willen zien waar hun element geplaatst is. Dit zou verbeterd kunnen worden door het systeembericht:

- 1. Een meer opvallende styling te geven.*
- 2. Dicht bij het geplaatste element weer te geven, zodat deze beter in het oog valt.*

Wanneer de oplossing niet zo voor de hand lag, heb ik geen verbetervoorstel gedaan. In deze gevallen moet er eerst een nieuw ontwerp worden gemaakt, dat vervolgens opnieuw getest kan worden.

Testrapport

Alle informatie die in deze paragraaf is beschreven, heb ik vastgelegd in een testrapport ([bijlage I](#)). In dit rapport kan zowel het volledige verslag van de test als de geanalyseerde data en de conclusies worden teruggelezen. Op deze manier kan de opdrachtgever, indien hij dat nodig acht, oordelen of mijn adviezen op de juiste manier zijn onderbouwd.

10. Nazorgfase

De nazorgfase was bedoeld om te zorgen dat de resultaten van mijn project op een bruikbare wijze kon worden overgedragen aan de opdrachtgever. Deze fase draaide volledig om het opstellen van een advies. Zo zou de opdrachtgever kunnen beoordelen welke ontwerpen hij wilde gebruiken in FontoXML, welke hij nog wilde verbeteren en welke hij niet wilde gebruiken.



10.1. Adviseren van de opdrachtgever

De allerlaatste projectactiviteit die ik heb uitgevoerd was het opstellen van een advies. De bedoeling was hier om de ontwerpen en bevindingen uit mijn project over te dragen aan de opdrachtgever. Op die manier kreeg de opdrachtgever een helder beeld van de resultaten van mijn project. Ook had hij zo een aantal heldere actiepunten tot zijn beschikking. Met deze punten kon dus worden bepaald wat voor vervolg er aan het project gegeven moest worden.

10.1.1. Totstandkoming van het advies

Allereerst heb ik een kleine samenvatting geschreven van de manier waarop mijn adviezen tot stand zijn gekomen. Hierin beschreef ik de belangrijkste activiteiten en producten uit het project. Zo kon de opdrachtgever beoordelen of hij vond dat het project goed verlopen was en de adviezen goed waren onderbouwd. Hierbij heb ik geprobeerd om dit stuk duidelijk en leesbaar te houden, zonder dat hiervoor andere documenten moesten worden geraadpleegd.

10.1.2. Advies op basis van het testrapport

Vervolgens heb ik het eerste deel van mijn adviezen opgesteld. Hierbij ging het om advies over het toepassen van de door mij ontworpen interfacepatronen. Dit waren de patronen die te maken hadden met de designproblemen voor de bewerking “element aanmaken”. Na voor deze ontwerpen een usabilitytest te hebben gehouden, kon ik een uitspraak doen over de gebruiksvriendelijkheid. Op deze manier heb ik, per designprobleem, geadviseerd wat de vervolgstappen zouden moeten zijn. Bij ieder advies legde ik nogmaals kort uit welke alternatieven er waren en waarom de desbetreffende keuze was gemaakt. In de meeste gevallen hield dit in dat er een van de twee ontwerpen moest worden geïmplementeerd, zoals bij het designprobleem “*omgaan met aanbevolen attributen bij het aanmaken van elementen*”, die te lezen is in de kadertekst hieronder.

Voor dit ontwerpprobleem is slechts een oplossing ontworpen. In deze oplossing wordt, bij het aanmaken van een element met een aangemoedigd attribuut, een popover getoond. In deze popover wordt de gebruiker verzocht om bepaalde actie uit te voeren. Hierbij kun je denken aan het invullen van een webadres of aan het aangeven van een bepaalde taal. De gebruiker kan dit verzoek afwijzen of er op ingaan, want er is geen sprake van een verplichte actie. Ongeacht of het attribuut wel of niet wordt ingevuld, het element wordt altijd geplaatst. Dit ontwerp is erg goed uit de gebruiksvriendelijkheidstest gekomen. Zonder dit te vragen, voerden alle gebruikers de aangemoedigde actie uit. ***Het advies is dan ook om dit ontwerp te implementeren in FontoXML.***

In sommige gevallen adviseerde ik om eerst een aantal verbetering toe te passen alvorens implementatie. Dit deed ik wanneer uit de usabilitytest was gebleken dat een ontwerp duidelijke tekortkomingen had op het gebied van gebruiksvriendelijkheid. Bij het ontwerp voor het “*aanmaken van elementen met een vaste volgorde*” bleek bijvoorbeeld dat deelnemers niet goed de situatie konden benoemen. Hierbij is dan ook het volgende advies opgesteld:

Voor het omgaan met het aanmaken van een element dat in een vaste volgorde moet staan zijn er ook twee opties ontworpen. Bij ontwerp A is de toolbarknop voor dit element altijd ingeschakeld en wordt het element automatisch op de juiste positie ingevoegd. In ontwerp B kunnen deze elementen worden ingevoegd met behulp van structuursuggesties in een zijpaneel. Het advies is hier **het ontwerp met de ingeschakelde toolbarknop door te ontwikkelen en vervolgens te implementeren**. De volgende argumenten zijn hiervoor te noemen:

1. Het zijpaneel bleek erg moeilijk vindbaar en lijkt niet te implementeren zonder een uitgebreide doorontwikkeling. De koppeling aan de elementen in het document is moeilijk te filteren en de structuursuggesties zouden daardoor voor alle elementen gedaan moeten worden.
2. Met ontwerp A kon de testtaak succesvol worden uitgevoerd: iets dat bij veel deelnemers met ontwerp B niet lukte.

Voor implementatie moet echter eerst nog een verbeterpunt worden behandeld. Er was namelijk maar een van de testdeelnemers die met zekerheid de situatie kon benoemen. Het systeembericht waarin deze situatie wordt verklaard, werd niet gelezen. Op basis van observaties bij de test lijkt het zo te zijn dat dit komt doordat gebruikers bij het aanmaken van elementen niet naar het hele scherm kijken. Omdat men zoekende is naar de plaats waar het nieuwe element wordt geplaatst, let niemand op de boven-of onderrand van het scherm. Voor beide posities is het systeembericht getest, en beide zonder succes. Aan te raden is dan ook om het systeembericht dicht bij het nieuwe element weer te geven. Wellicht zelfs in het document. Hierbij moet wel worden voorkomen dat het bericht opdringerig wordt: een verklaring moet nooit als storend worden ervaren.

10.1.3. Advies op basis van het onderzoeksrapport

Het tweede deel van mijn adviezen heb ik gebaseerd op het onderzoeksrapport. Niet alle designproblemen die uit dit rapport naar voren kwamen, zijn in mijn project behandeld. Om het probleem om te lossen dat aan de basis stond van dit project, zou dit wel moeten gebeuren. Om die reden heb ik in mijn advies opgenomen dat de overige designproblemen nog moeten worden

uitgewerkt. Alleen dan kan worden vastgesteld welke interfacepatronen er nodig zijn voor de bewerking van een element.

De adviezen en de totstandkoming daarvan heb ik samengevoegd in een adviesrapport, gericht op de opdrachtgever. Dit adviesrapport is opgenomen in de externe bijlagen als bijlage K.

Deel 3

Evaluatie

In de evaluatie kijk ik terug op het project en beoordeel ik hoe het gegaan is. Daarvoor begin ik met de procesevaluatie. Hierin beschrijf ik vooral welke activiteiten ik verkeerd heb uitgevoerd, maar ook welke activiteiten ik op het verkeerde moment heb uitgevoerd. Dit is vooral bedoeld om te voorkomen dat ik dezelfde fouten in de toekomst opnieuw zal maken. In de productevaluatie beoordeel ik de kwaliteit van mijn (deel)producten. Hierbij kijk ik of deze producten in hun doel hebben voldaan en waar dit eventueel beter had gekund. Tot slot kijk ik terug op het wel of niet op niveau behalen van de competenties die ik bij dit project wilde bewijzen.



11. Procesevaluatie

11.1. Behalen van projectdoelen

Zoals eerder genoemd is dit project in het leven geroepen vanwege het volgende probleem: *“Het kost momenteel veel tijd om voor het bewerken van de verschillende elementen in een XML-schema de juiste interface te kiezen”*. Bij dit probleem stelde ik samen met het team van FontoXML de volgende doelstelling op: *“op een voorspelbare manier de interface van de applicatie kunnen configureren bij een implementatie met een nieuw XML-schema”*. Die doelstelling is na afloop van dit project slechts deels behaald. Tijdens de onderzoeksfase is een aantal factoren in kaart gebracht dat invloed heeft op het bewerken van XML-elementen. Ook is in kaart gebracht op welke bewerkingen deze factoren invloed hebben.

Omdat er echter niet voor alle bewerkingen interfacepatronen zijn ontworpen, kan de interface nog niet zomaar geconfigureerd worden. Voor de bewerking “element aanmaken” zou dit kunnen. Hier zijn voor alle factoren van invloed interfacepatronen ontworpen die kunnen worden toegepast in FontoXML. Om de volledige interface van een element te configureren zullen ook de overige bewerkingen moeten ontworpen. Er zullen dus nog een of meerdere vervolgetrajecten moeten komen voordat dit mogelijk is. Het bovenliggende doel van dit project (*het hebben van een gebruiksvriendelijke XML-editor voor vakexperts*) is wel een stukje dichterbij gekomen. Dit project was niet bedoeld om dit doel volledig te behalen, maar om hier aan bij te dragen. Door de geteste interfacepatronen toe te passen binnen FontoXML zal het aanmaken van XML-elementen gebruiksvriendelijker worden.

11.2. Algemene projectaanpak

Over het algemeen is mijn project naar behoren verlopen. De onderzoeksfase heeft geleid tot een overzicht van bewerkingen en factoren dat een goede basis vormt voor het ontwerpen en vervolgens kiezen van de juiste interfacepatronen. De onderzoeksfase heeft echter wel erg veel tijd gekost, waardoor de hoeveelheid ontworpen interfacepatronen redelijk beperkt is gebleven. Ondanks dat dit voor de opdrachtgever geen probleem was, vond ik dit zelf wel jammer. Ik koos er vooraf voor om het onderzoek aan een stuk door uit te voeren, zodat ik een compleet product had voordat ik met het ontwerpen aan de slag zou gaan. Dit was een van de hoofdredenen dat ik heb gekozen voor een watervalmethode. Hierdoor kon ik de uitloop in de onderzoeksfase niet optimaal verwerken.

Met de kennis die ik nu heb, zie ik toch manieren waarop dit project op een Agile manier had kunnen uitgevoerd. Door het onderzoek op te splitsen in losse, apart uitvoerbare onderdelen, had ik sneller tot de ontwerpfase kunnen komen. Vervolgens had ik, samen met opdrachtgever, per sprint

kunnen bepalen of het prioriteit had om meer *onderzoek* of meer *ontwerp* te gaan uitvoeren. Zo had ik bijvoorbeeld eerst tot een lijst van vier bewerkingen kunnen komen, vervolgens tot drie factoren van invloed en hiermee het begin van de tabel kunnen maken. Voor deze tabel had ik dan al kunnen ontwerpen en prototypen. Daarna zou ik weer een stukje onderzoek kunnen uitvoeren, waarmee de tabel vervolgens weer kon worden uitgebreid. Dan konden er weer nieuwe ontwerpen worden gemaakt en kreeg het prototype weer meer functionaliteit. Zo lever je constant een bruikbaar product, dat telkens uitgebreider wordt.

Deze alternatieve aanpak voelt wat minder gecontroleerd dan de vaste projectaanpak die ik nu heb gekozen, maar was ook een stuk flexibeler geweest. Ik ben niet van mening dat deze nieuwe aanpak per se beter is, maar ik zie het nu wel als een reële optie. Dat was bij aanvang van mijn project totaal niet het geval.

11.3. Planning

De planning is, zoals meestal, een behulpzaam product gebleken bij het uitvoeren van dit project. Aan de hand van de planning kon ik overzicht bewaren en soms een tandje bijsteken waar nodig. Toch ben ik ook (zoals meestal) een aantal keer afgeweken van de planning. Zo kostte het onderzoek naar XML-bewerking meer tijd dan verwacht, waardoor uitloop ontstond. Deze werd weer opgevangen met het vervallen van een aantal activiteiten, zoals de User Scenario's (in de planning nog verkeerd geformuleerd als use cases). Een ander stukje van de uitloop heb ik opgevangen in de avonduren en de weekenden. Ik hield mijn planning bij in de webapplicatie Trello, waarin je met kaartjes verschillende activiteiten kunt bijhouden. Op deze manier kon ik duidelijk per dag bijhouden wat ik moest doen en of dat allemaal lukte.

11.4. Taakanalyse

De taakanalyses had ik aanvankelijk in het begin van de ontwerpfase ingepland. Eenmaal begonnen aan het project besloot ik om deze naar voren te halen: bij het in kaart brengen van de bewerkingen leek mij dit een handig hulpmiddel. Ik analyseerde enkel de "kale" bewerking en hield nog geen rekening met een eventuele interface. Dit deed ik bewust, omdat ik het nog geen ontwerpkeuzes wilde maken. Bij het uitwerken van deze analyses bleken zij echter maar heel "klein" te zijn: meer dan vier stappen kwamen er vaak niet voor.

De taakanalyses hadden hun functie beter kunnen vervullen als ik deze toch later in het project, pas in de ontwerpfase, zou hebben opgesteld. Wanneer ik deze taakanalyses *tijdens* het ontwerpen had opgesteld, had ik veel beter rekening kunnen houden met de verschillende stappen en routes die mogelijk waren. Ik ben hier dus afgeweken van mijn planning, terwijl ik dat eigenlijk niet had moeten doen.

11.5. Onderzoeken van Best Practices

Een van factoren die invloed bleek te hebben op de bewerking van elementen, was “best practices in XML-bewerking”. Na dit ontdekt te hebben ben ik intensief deskresearch gaan doen om te achterhalen om welke best practices het hier ging. Dit leverde weinig resultaat op, kostte erg veel tijd en veroorzaakte zelfs uitloop op de planning. Uiteindelijk vond ik een korte lijst met mogelijke best practices. Op dat moment stapte ik naar het team om te overleggen wat zij van deze lijst vonden.

Het was beter geweest als ik een stuk eerder naar het team zou zijn gestapt. Niet alleen bleek dat het enige(!) resultaat van mijn deskresearch onbruikbaar was, maar ook bleek dat het antwoord heel eenvoudig bij het team te vinden was. Er bleek namelijk een aantal best practices te zijn die zij zelf al hadden bedacht. Het was de bedoeling dat deze zouden worden meegenomen in het onderzoek. Ik heb hier dus te hard geprobeerd om zelf tot een resultaat te komen, zonder hulp van anderen. Ik had beter bij ieder deel van het onderzoek kunnen beginnen bij het team: zij hadden immers al een hoop ervaring en kennis met het betreffende onderwerp.

11.6. Onderzoeken van gebruikers

Het gebruikersonderzoek bij dit project is, vergeleken bij mijn eerdere ervaringen, erg goed verlopen. Omdat het contact van Liones met de klant al erg goed was, kostte het weinig moeite om afspraken te maken. Het was zelfs mogelijk om, zonder vergoeding, tijdens werktijd met een aantal gebruikers af te spreken. Sommige van deze gebruikers waren wat zenuwachtig, maar doordat ik mij daar vooraf goed op had voorbereid, kon ik hen makkelijk geruststellen. De combinatie interview en observatie is mij ook zeer goed bevallen. De resultaten hiervan leerden mij echt wat over de manier waarop gebruikers hun werk uitvoerden.

Er was nog wel een verbeterpunt. Tijdens het observeren heb ik de gebruikers vrijgelaten in het “laten zien van hun dagelijkse werk”. Hierdoor waren de resultaten niet altijd relevant voor mijn project: ik was heel specifiek bezig met bepaalde taken. Het was dus beter geweest om de gebruikers hier te vragen deze taken uit te voeren. Zo had ik resultaten kunnen verzamelen die nog beter toepasbaar op mijn project waren geweest.

Wat betreft de contextual inquiries was het wellicht beter geweest om deze uit te voeren *na* het uitwerken en prioriteren van de bewerkingen. Op deze manier had ik mij meer kunnen focussen op het observeren van de bewerking “aanmaken” om te zien hoe deze nu meestal wordt uitgevoerd. Anderzijds was er dan een hoop algemene informatie over de context van het gebruik niet gevonden. Dit had weer een negatieve invloed gehad op het grote geheel van dit project.

11.7. User Scenario's

Een van de activiteiten die ik voor dit project had ingepland was het opstellen van User Scenario's. De bedoeling hiervan was om meer inzicht te creëren in de context van de verschillende bewerkingen. Toen ik eenmaal bij dit punt van het project was aanbeland (het onderzoek naar XML-bewerkingen was dus helemaal afgerond), was het opstellen van scenario's naar mijn idee niet meer erg relevant. Waar ik bij aanvang van het project nog dacht dat deze scenario's belangrijk waren voor het begrijpen van de situatie van gebruik, was mijn mening nu veranderd. Doordat ik bij alle designproblemen al een aantal voorbeelden van elementen had opgezocht, waren er al genoeg realistische voorbeelden om mee te ontwerpen en te testen. Om die reden heb ik het opstellen van scenario's geschrapt uit de planning.

11.8. Benchmark

De benchmark heeft een aantal bruikbare interfacepatronen opgeleverd, waarop ik later een aantal ontwerpen heb kunnen baseren. Deze activiteit had waarschijnlijk echter nog een stuk meer vruchten af kunnen werpen. De applicaties die ik koos om te analyseren in mijn benchmark waren Google Docs, Microsoft Word, OxygenXML en EasyDita. Waar de laatste twee veel resultaten opleverden, bleef dit bij de eerste twee echter uit. Aanvankelijk koos ik Docs en Word omdat dit applicaties zijn waarvan de gebruikers gewend was om ze te gebruiken. Voor eventuele interfacepatronen die ik hier zou vinden, zou de kans dus groter zijn dat gebruikers al bekend zijn met de werking hiervan.

Doordat er in deze applicaties echter geen XML bewerkt kan worden, bleek dat nagenoeg geen van de designproblemen die ik in mijn benchmark onderzocht in deze applicaties kon worden gereproduceerd. Er kon hierdoor dus ook niet gekeken worden naar de oplossingen die er werden toegepast. Ik herkende dit probleem pas halverwege het uitvoeren van de benchmark en besloot vervolgens om toch door te zetten. Waar ik het probleem niet kon reproduceren ben ik gaan zoeken naar vergelijkbare situaties, om toch een resultaat te krijgen. Dit kostte veel tijd en leverde weinig op. Achteraf was het hier beter geweest om wat tijd te steken in het zoeken van applicaties die wel met deze designproblemen om moesten gaan. Op deze manier had de benchmark hoogstwaarschijnlijk meer bruikbaar resultaat opgeleverd.

11.9. Onderzoek naar bestaande interfacepatronen

Aanvankelijk had ik in mijn planning een activiteit ingepland waarbij ik onderzoek zou doen naar bestaande interfacepatronen. Hierbij wilde ik op zoek gaan naar veelgebruikte patronen voor het soort designproblemen waar ik in dit project mee te maken had. Eenmaal aangekomen op dit punt in het project, bleek dit echter geen activiteit die ik heel serieus aan wilde pakken. Ik vond het veel logischer om een aantal concepten te gaan bedenken en daarbij in literatuur en websites zoeken

naar ideeën, zoals beschreven in hoofdstuk 7.2.1. Er is dan ook geen concreet product uit deze activiteit voortgekomen. Eerder heeft het de basis gevormd voor een aantal concepten die ik ontworpen heb. Doordat deze activiteit opging in het bedenken van concepten, is er iets afgeweken van de planning. Achteraf lees ik ook nergens het product terug dat ik met deze activiteit had willen maken. Dit is een activiteit die voorafgaand aan mijn project kennelijk logisch aanvoelde, maar tijdens het project niet meer.

11.10. Voorbereiden van de usabilitytest

Over mijn voorbereiding op de usabilitytest ben ik achteraf zeer tevreden. Het boek dat ik hierbij als naslag heb gebruikt bood zeer veel informatie, waardoor mijn testplan completer is geworden dan de testplannen die ik het verleden heb opgesteld. Ook ging het werven van de testpersonen, net als bij het gebruikersonderzoek, weer erg goed. Dit had echter voornamelijk te maken met de welwillendheid van de klant. Dit bleek dan ook zeer belangrijk te zijn voor het kunnen uitvoeren van goede tests.

Achteraf had de kwaliteit van de aandachtspunten die ik heb opgesteld wat hoger gekund. Hierdoor was het achteraf soms wat moeilijk om de gebruiksvriendelijkheid van een ontwerp te “meten”. Ondanks het feit dat ik bewust heb gekozen voor wat ik “kwalitatieve meetpunten” noem, waren de resultaten soms erg moeilijk te analyseren. Toch konden er in de meeste gevallen wel conclusies worden getrokken uit deze meetpunten. In de toekomst wil ik echter proberen om meer meetbare punten op te stellen, zodat ook in cijfers is aan te tonen hoe succesvol een ontwerp is. Ik zal dan gaan onderzoeken of factoren zoals doorlooptijd of een aantal foute clicks meetpunten voor gebruiksvriendelijkheid kunnen zijn. In dat geval kunnen er wellicht ook doelen worden opgesteld. Je kunt dan bijvoorbeeld denken aan een maximale doorlooptijd van 3 minuten voor een bepaalde taak.

11.11. Uitvoeren van de usabilitytest

Over het algemeen is het uitvoeren van de usabilitytest goed verlopen. Sommige deelnemers waren vooraf nog wat zenuwachtig, maar omdat ik daar op had gerekend kon ik hen vrij gemakkelijk geruststellen door de druk weg te nemen. Hierdoor zijn de testtaken naar mijn idee op een vrij natuurlijke manier doorlopen: iets wat de kwaliteit van de resultaten verhoogd.

Een punt dat iets minder goed ging bij de uitvoer van de test was het letten op de aandachtspunten. Bij een aantal taken ben ik niet actief bezig geweest met het verzamelen van antwoorden bij deze punten. Hierdoor vergat ik sommige vragen die ik aan de deelnemers had willen stellen na het uitvoeren van een taken. Dit kwam vooral doordat ik soms te actief bezig was met het observeren van het scherm, waardoor ik het overzicht zo nu en dan verloor. Gelukkig is dit beperkt gebleven tot drie keer, maar alsnog is het zonde van de resultaten.

Verder was het erg prettig dat ik beschikking had over een tweede observant. Nadat ik achteraf de aantekeningen ontving die hij tijdens de test had gemaakt, merkte ik dat hier voor mij best wat nieuwe dingen in stonden. Zo kwam ik dus aan resultaten die ik anders waarschijnlijk had gemist.

12. Productevaluatie

12.1. Onderzoeksrapport (eindproduct)

Het onderzoeksrapport dat ik heb opgesteld is een goed leesbaar product geworden met een duidelijke verhaallijn. Vooral de conclusies, met daarin de verbandentabel, is volgens mijn bedrijfsmentor een nuttig product voor Liones geworden. In de rest van het rapport kan gelezen worden hoe deze tabel tot stand is gekomen en dus worden gecontroleerd of deze aanpak klopt.

Door in het eerst deel van het rapport de achtergrond van het onderzoek uit te leggen, zou deze wellicht ook nog bruikbaar kunnen zijn voor andere partijen die met soortgelijke problemen te maken hebben, indien Liones besluit dat dit is toegestaan.

12.2. Benchmark

Het rapport van de benchmark is naar mijn mening wat te rommelig opgemaakt. De afbeeldingen verstoren de lay-out en de toon van schrijven is te inconsistent. Verder had er in de conclusies wellicht nog een tabel kunnen worden opgenomen waarin de scores van de verschillende applicaties per designprobleem naast elkaar worden gezet. Op deze manier was er een visueel betere onderbouwing ontstaan voor de conclusies die ik heb getrokken.

12.3. Ontwerprapport (eindproduct)

Het ontwerprapport was bedoeld om naar de opdrachtgever te communiceren welke oplossingen ik heb ontworpen en hoe ik tot die oplossingen ben gekomen. De interactieontwerpen waren hierbij het belangrijkste onderdeel.

Als eindproduct voor het interactieontwerp heb ik gekozen voor handgemaakte storyboards met een tekstuele onderbouwing. Deze combinatie bleek voor dit project een gouden keuze: het maken van de schetsen kostte relatief weinig tijd en in combinatie met de tekstuele onderbouwing was voor zowel de opdrachtgever als voor mijzelf goed te begrijpen hoe een interactie moest werken. Deze onderbouwing bleek echter wel cruciaal: het blijft naar mijn idee nagenoeg onmogelijk om alle aspecten van een interactief ontwerp uit te werken op papier. Tekst bleek hierbij een snelle, begrijpelijke manier om een schets of storyboard completeren.

De “visuele kwaliteit” van mijn schetswerk is niet erg hoog. Lijnen zijn met de hand getrokken en mijn handschrift laat soms ook te wensen over. Als dit project was uitgevoerd voor een ander type klant (lees: geen internetbureau) had ik mijn schetsen waarschijnlijk nog digitaal uitgewerkt. Dit oogt professioneler en strakker, wat nodig zou zijn om een goed beeld bij de klant te vormen. Omdat mijn “klant” echter ook mijn opdrachtgever was, bleek dit inderdaad niet nodig. Ook

wanneer een externe vormgever of programmeur nog met mijn ontwerpen zou moeten werken, had ik deze waarschijnlijk digitaal uitgewerkt.

Het team van Liones had echter al erg veel ervaring met het ontwerpen van websites en schetsen. Hierdoor waren mijn schetsen voor hen goed te begrijpen. Ook zagen zij in dat het omzetten van papieren schetsen naar digitale schetsen een tijdrovend proces zou zijn, waarvan de uitkomst weinig meerwaarde zou leveren voor het project. De schetsen waren in deze vorm namelijk een prima basis voor overleg en voor het maken van de prototypes.

Het enige verbeterpunt in mijn schetsen is dat ik nóg consistentere had kunnen zijn in de stijl. Nu is er soms wat afwijking in het type nummering dat ik gebruik. Ook variëren soms de kleuren die ik heb gebruikt voor het toepassen van nadruk. Dit moet in de toekomst nog strakker bijgehouden worden, maar zal ook automatisch verbeteren naarmate ik langer met dezelfde stijl werk.

12.4. Prototype

Over het uiteindelijke prototype ben ik behoorlijk tevreden. De interacties zoals die in het prototype zijn opgenomen komen zeer sterk overeen met de interacties die zijn uitgewerkt in het ontwerprapport. Hier heb ik eigenlijk geen concessies hoeven doen door bijvoorbeeld een beperking in mijn programmeervaardigheden. Het prototype voelt ook echt als een werkende tekstverwerker. Daarnaast is er een aantal functionaliteiten, zoals de focus op een bepaald topic of de werking van de toolbar, die ik in zeer weinig tijd heb kunnen nabouwen van FontoXML. Hierdoor lijkt het prototype op sommige vlakken enorm veel op de echte applicatie.

In deze representativiteit ligt naar mijn idee ook de grootste zwakte van het prototype. Logischerwijs bevat mijn prototype niet alle functionaliteiten van FontoXML: het is niet mogelijk om in een paar weken een applicatie na te bouwen waar een team van twaalf man al een jaar aan werkt. Toch was het wel prettig geweest als mijn ontwerpen in een “echte” omgeving getest hadden kunnen worden. Ik ben er namelijk van overtuigd dat de gebruikerstesten op deze manier niet alleen feedback over mijn ontwerpen, maar ook over de rest van de applicatie hadden opgeleverd. Zo waren de tests voor FontoXML dus nóg waardevoller geweest. Ik had dan een aantal “randadviezen” kunnen meegeven.

12.5. Adviesrapport (eindproduct)

Het adviesrapport is zeer beknopt gebleven. Er is kort de lezen hoe het advies tot stand is gekomen en wat de adviezen precies zijn. Op deze manier is het document goed leesbaar gebleven, zelfs zonder dat er een managementsamenvatting nodig was.

13. Behalen van competenties

In het laatste hoofdstuk van dit deel evalueer ik het behalen van competenties. Voorafgaand aan dit project heb ik, in overleg met mijn eerste examiner en mijn bedrijfsmentor, een vijftal competenties opgesteld. Voor deze competenties wilde ik tijdens het uitvoeren van dit project bewijzen dat ik ze op een hbo-waardig niveau kon uitvoeren. In dit hoofdstuk beschrijf ik of dat naar mijn mening is gelukt en waarom (niet).

13.1. Opstellen plan van aanpak

Het opstellen van een plan van aanpak (PVA) is een activiteit die ik tijdens mijn opleiding menig maal heb moeten uitvoeren. Met behulp van mijn ervaring en het boek van Roel Grit heb ik vooraf een zinnige selectie kunnen maken uit hoofdstukken die voor dit project wel en niet van belang waren. Het plan van aanpak was hierdoor een zinnig hulpmiddel bij het bewaken van de vorderingen in het project en heeft hierdoor zeker in zijn doel voorzien.

13.2. Opstellen interaction design

Het ontwerpen van interaction design vormde de kern van dit project. Een goed interactieontwerp is gebaseerd op onderzoek: er moet kunnen worden bewezen dat het ontwerp gebruiksvriendelijk is. Ook vooraf is zoveel mogelijk kwaliteitsgarantie al gewenst. Het is dus belangrijk om alleen met concepten aan de slag te gaan waarvan al eerder is bewezen dat deze werken voor een soortgelijk designprobleem. Of, met concepten waarvan zowel de klant als de gebruiker en de ontwerper denken dat deze succesvol zouden kunnen zijn. Door mijn ontwerpen te baseren op patronen uit de benchmark en vakliteratuur heb ik een groot deel van deze kwaliteit gewaarborgd. Eventuele nieuwe concepten heb ik veelvuldig besproken met het team van FontoXML. Op deze manier had ik vooraf zoveel mogelijk zekerheid dat ik de goede weg zat. Door deze patronen ook nog eens achteraf te testen met representatieve gebruikers, denk ik dat de kwaliteit van mijn ontwerpen een goede onderbouwing heeft.

13.3. Uitwerken prototype

In het prototype kwamen de ontwerpen tot leven. Waar het ontwerprapport bestaat uit stilstaande, papieren schetsen, konden de interacties in het prototype daadwerkelijk worden uitgevoerd. Door te kiezen voor een HTML-prototype had ik hierin alle vrijheid. Door mijn eerdere ervaringen met Javascript kon ik deze goed toepassen en mijn ontwerpen goed vertalen naar de code. Het resultaat was een replica van FontoXML waarbij de leek het onderscheid maar moeilijk zou kunnen vinden. Dit leverde dan ook een goede basis voor de tests.

13.4. Opstellen testplan & testrapportage

Voor dit project heb ik een testplan uitgewerkt met meer diepgang en doelgerichtheid dan ooit tevoren in mijn opleiding. Dit is vooral te danken aan het zeer uitgebreide handboek dat ik hiervoor gevonden heb (Rubin, 2008). Ook hier heb ik echter nog bewust een selectie gemaakt de onderdelen die wel of niet van belang waren voor dit project. Het testplan was voor mijn bedrijfsmentor zeer helder en leesbaar en beschrijft alle aspecten waar bij het uitvoeren van de test rekening mee gehouden moest worden.

Bij het opstellen van het testrapport was het voor mij belangrijk dat deze beknopt en snel te lezen was. Om die reden heb ik alle samengevatte resultaten en overzichten als kern gebruikt. De volledige uitwerking van de testresultaten heb ik in de bijlages opgenomen. Zo kon er snel tot de conclusies van de tests worden gekomen en bleef het testrapport vooral een basis voor het adviesrapport. Dit was immers de reden waarvoor ik het testrapport wilde maken.

13.5. Usabilitytest opzetten en uitvoeren

Mede dankzij de medewerking vanuit ThiemeMeulenhoff is de test erg goed verlopen. Het contact vooraf verliep soepel, er konden snel duidelijke afspraken worden gemaakt en deze werden goed nagekomen. Het is gelukt om te testen met representatieve testdeelnemers: vakexperts die in de toekomst FontoXML daadwerkelijk zullen gaan gebruiken. Ondanks het feit dat ik maar met vier personen heb kunnen testen heeft dit toch een hoop bruikbare resultaten opgeleverd. Ik heb zowel over de bruikbaarheid van de ontworpen oplossingen als over mogelijke verbeterpunten goed onderbouwde adviezen kunnen uitbrengen. Ook is wederom benadrukt hoe mensen allemaal verschillend kunnen denken. Sommige ontwerpen die voor mij volledig logisch leken, werden door de deelnemers totaal anders ervaren. Hierdoor heb ik de usabilitytest als een van de meest nuttige activiteiten ervaren. Ik ben dan ook zeer tevreden over de resultaten.

Literatuurlijst

Gedrukte bronnen

Buxton, W. (2007). *Sketching user experiences: Getting the design right and the right design*(pp. 136). Amsterdam: Elsevier/Morgan Kaufmann.

Grit, R. (2011). Het Project. In *Projectmanagement* (6e ed.). Groningen/Houten: Noordhoff Uitgevers.

Hall, E (2013). *Just enough research*(pp. 82-84, 90). New York: Abookapart.

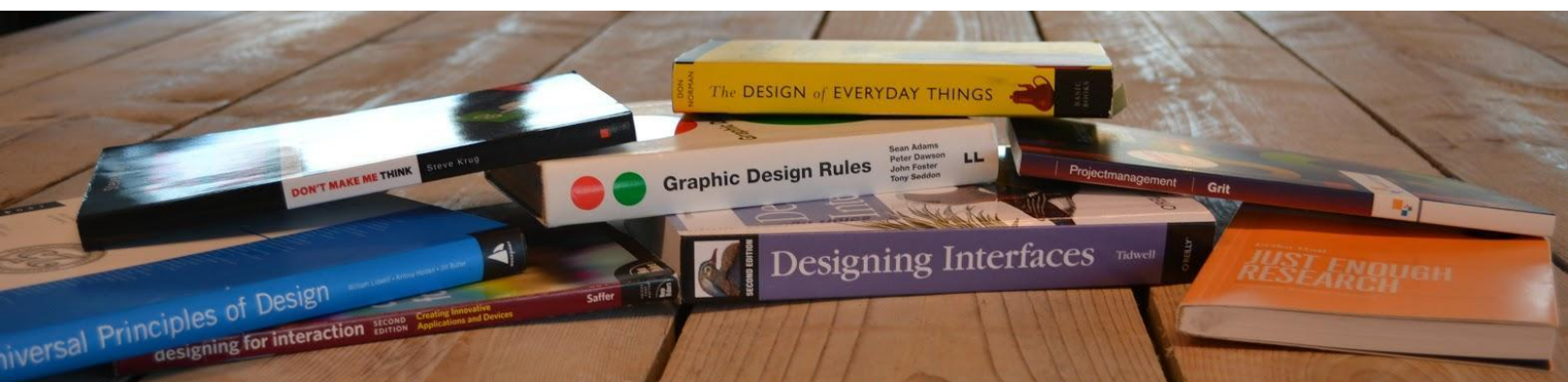
Kreitzberg, C. (2008). *The LUCID Framework - An Introduction*. New Jersey: Cognetics Corporation.

Krug, S. (2011). *Don't make me think: een nuchtere kijk op webusability* (2e ed., pp. 121-143). Zaltbommel: Thema.

Rubin, J., & Chisnell, D. (2008). *Handbook of usability testing: how to plan, design, and conduct effective tests* (2e ed.). Indianapolis: Wiley Pub.

Saffer, D, (2010). *Designing for interaction*(pp.33-35). Berkeley: New riders.

Tidwell, J. (2010). *Designing interfaces* (2e ed., pp. 97-100). Sebastopol: O'Reilly Media.



Digitale bronnen

Amazon Best Sellers. (2014, 4 oktober). Opgehaald van <http://www.amazon.com/Best-Sellers-Software-Word-Processing/zgbs/software/22954>

Burnard, L., & Bauman, S. (2014, 16 september). *TEI Guidelines*. Opgehaald van <http://www.tei-c.org/release/doc/tei-p5-doc/en/html/>

Cerejo, L. (2010, 16 juni). *Design Better And Faster With Rapid Prototyping*. Opgehaald van <http://www.smashingmagazine.com/2010/06/16/design-better-faster-with-rapid-prototyping/>

Eberlein, K. (2010, 1 december). *Darwin Information Typing Architecture (DITA) Version 1.2*. Opgehaald van <http://docs.oasis-open.org/dita/v1.2/os/spec/DITA1.2-spec.html>

Hornsby, P. (2010, 8 februari). *Hierarchical Task Analysis*. Opgehaald van <http://www.uxmatters.com/mt/archives/2010/02/hierarchical-task-analysis.php>.

International Organization for Standardization. (2010). *ISO IS 9241 deel 210: Human-centred design for interactive systems*. Genève, Zwitserland

Linowski, J. (2012, 23 april). *Interactive Sketching Notation*. Opgehaald van http://www.linowski.ca/downloads/ISN_1.2_Introduction.pdf

Nielsen, J. (1995, 1 januari). *10 Usability Heuristics for User Interface Design*. Opgehaald van <http://www.nngroup.com/articles/ten-usability-heuristics/>

Nielsen, J. (2007, 27 mei). *The Myth of the Genius Designer*. Opgehaald van <http://www.nngroup.com/articles/the-myth-of-the-genius-designer/>

Nielsen, J. (2012, 16 januari). *Thinking Aloud: The #1 Usability Tool*. Opgehaald van <http://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/>

Savourel, Y., & Kosek, J. (2008, February 13). *Best Practices for XML Internationalization*. Opgehaald van <http://www.w3.org/TR/xml-i18n-bp/#AuthoringTime>

Spool, J. *5 Design Decision Styles. What's Yours?* (2009, 21 januari). Opgehaald van http://www.uie.com/articles/five_design_decision_styles

Spool, J. (2005, 15 februari). *Seven Common Usability Testing Mistakes*. Retrieved from https://www.uie.com/articles/usability_testing_mistakes/

Spool, J. (2011, 25 maart). *The Unintuitive Nature of Creating Intuitive Designs*. Opgehaald van <https://vimeo.com/21651127>

Thompson, H. (2004, 28 oktober). *XML Schema Part 1: Structures Second Edition*. Opgehaald van <http://www.w3.org/TR/xmlschema-1/>

UXPA. *What is User-Centered Design?* (z.j.). Opgehaald van http://www.usabilityprofessionals.org/usability_resources/about_usability/what_is_ucd.html

W3C. *Notes on User Centered Design (UCD)*. (2004, 31 maart). Opgehaald van <http://www.w3.org/WAI/redesign/ucd>

Wikipedia. (z.j.). *MoSCoW-methode*. Opgehaald van <http://nl.wikipedia.org/wiki/MoSCoW-methode>

Note

Op 23 december 2014 ontdek ik dat de links naar de website www.usabilityprofessionals.org down zijn. Op de nieuwe site van het UXPA zijn deze artikelen niet meer terug te vinden. Ik heb hen die dag een email gestuurd met een verzoek om alternatieven. Tot op 5 januari 2015 is daar nog niet op gereageerd.

Verklarende woordenlijst

Constraints

Vooraf bepaalde en vastgelegde voorwaarden waaraan het project in ieder geval moet voldoen.

Kind-element

Kortweg: kind

Logischerwijs is er dus ook sprake van een kind-element: een element dat direct in een ander element staat. In het voorbeeld hieronder zou” `<quote>` het kind van de `<paragraph>` zijn.

```
<paragraph>  
  <quote>Ik ben een kind, dus heb ik een ouder.</quote>  
</paragraph>
```

Ouder-element

Kortweg: ouder

Een ouder-element is het element dat direct om een bepaald element heen staat. Hieronder volgt hier een voorbeeld waarbij de `<paragraph>` de ouder is van `<quote>`.

```
<paragraph>  
  <quote>Ik ben een kind, dus heb ik een ouder.</quote>  
</paragraph>
```

(XML)Attribuut

Attributen zijn eigenschappen van elementen die kunnen worden vastgelegd in XML. Een attribuut druk je uit door deze te plaatsen binnen de opening van een element, na de elementnaam. Attributen kunnen ook een waarde bevatten. Hier een voorbeeld waarbij “keyword1” de waarde van het “id”-attribuut is: `<keyword id="keyword1">usability</keyword>`.

(XML-)Element

Ieder onderdeel van een tekst dat als dusdanig moet worden kunnen worden onderscheiden, wordt een apart element. Elementen worden in XML vastgelegd door een woord te plaatsen in angle brackets, zoals in het volgende voorbeeld: `<keyword>usability</keyword>` Zoals hier ook te zien is hebben de meeste elementen een begin en een einde. Een element kan zo groot zijn als een document (`<topic>`) of zo klein als een steekwoord (`<keyword>`).

(XML)-Schema

kortweg: schema

In een schema worden regels vastgelegd waaraan een XML-document moet voldoen om “valide” te zijn. Deze regels kunnen op verschillende eigenschappen en relaties van elementen gebaseerd zijn.

User Interface

kortweg: UI

De gebruikersinterface omvat alle aspecten van een product die faciliteren in de interactie tussen de gebruiker en dat product. Bij digitale producten bestaat dit dus vooral uit verschillende elementen op het scherm, maar ook het toetsenbord en de muis zouden onder deze interface kunnen worden genoemd.