



Bijlagen

Optimalisering en automatisering van interne foutafhandeling procedures

De integratie van een module waarin fouten op websites worden afgehandeld en getraceerd naar de oorsprong

Geschreven door: E. Soek
Studentnummer: 11004010

Soort document: Afstudeerverslag
Examinator 1: A.M.J.J. Lousberg – Orbons
Examinator 2: E.M. van Doorn

Onderwijsinstelling: De Haagse Hogeschool
Faculteit: IT&Design
Opleiding: Informatica

In uitvoering voor: Floro Webdevelopment B.V.

Rotterdam, februari 2016

Versie 1.0

Inhoudsopgave

Bijlage A: Afstudeerplan	1
Bijlage B: Plan van aanpak.....	4
Bijlage C: Requirementsrapport pre herziening.....	15
Bijlage D: Toolselectie	25
Bijlage E: Requirementsrapport na herziening	35

Bijlage A: Afstudeerplan

Informatie afstudeerder en gastbedrijf

Afstudeerblok: 2016-1.1 (start uiterlijk 8 februari 2016)

Startdatum uitvoering afstudeeropdracht: 8 februari 2016

Inleverdatum afstudeerdossier volgens jaarrooster: 3 juni 2016

Studentnummer: 11004010

Achternaam: mw Soek

Voorletters: E

Roepnaam: Eva

Adres: Foulkeslaan 9

Postcode: 2625 PW

Woonplaats: Delft

Telefoonnummer: 015-2560078

Mobiel nummer: 06-14048862

Privé emailadres: emhsoek@gmail.com

Opleiding: Informatica

Locatie: Den Haag

Variant: voltijd

Naam studieloopbaanbegeleider: J.C.M. Wieman

Naam begeleidend examiner: A.M.J.J. Lousberg - Orbons

Naam tweede examiner: E.M. van Doorn

Naam bedrijf: Floro Webdevelopment

Afdeling bedrijf: Development

Bezoekadres bedrijf: Westblaak 7R

Postcode bezoekadres: 3012 KC

Postbusnummer: -

Postcode postbusnummer: -

Plaats: Rotterdam

Telefoon bedrijf: 010-2143283

Telefax bedrijf: -

Internetsite bedrijf: www.floro.nl

Achternaam opdrachtgever: dhr Dorst

Voorletters opdrachtgever: S

Titulatuur opdrachtgever: -

Functie opdrachtgever: Project manager

Doorkiesnummer opdrachtgever: -

Email opdrachtgever: stephan@floro.nl

Achternaam bedrijfsmentor: dhr Dorst

Voorletters bedrijfsmentor: S

Titulatuur bedrijfsmentor: -

Functie bedrijfsmentor: Project manager

Doorkiesnummer bedrijfsmentor: -

Email bedrijfsmentor: stephan@floro.nl

Doorkiesnummer afstudeerder: -

Functie afstudeerder (deeltijd/duaal): -

Titel afstudeeropdracht: Het uitvoeren van een toolselectie en het ontwerpen en bouwen van een geautomatiseerd testsysteem.

Opdrachtomschrijving

1. Bedrijf

Floro Webdevelopment B.V. is ontstaan uit het samengaan van Van Leerdam Websolutions en Robin van Holst Webdesign op 1 januari 2010. Vanaf dat moment werd het bedrijf gerund door twee partners: Floris en Robin. Floro is een bedrijf dat zich richt op webdevelopment van websites, webshops en cloudoplossingen. De producten en diensten die het bedrijf levert, worden ondersteund door hun eigen informatiesystemen. Het bedrijf telt ongeveer 25 werknemers.

2. Probleemstelling

Floro heeft een framework ontwikkeld, dat iedere website die het bedrijf maakt ondersteunt. Dit framework kan als motor worden gezien van alle opdrachten van het bedrijf. Het is dus erg belangrijk dat dit framework te allen tijde optimaal presteert. Op het moment moet Floro na wijzigingen in het framework alle websites testen om te kijken of deze nog werken. Dit is erg onhandig en dit willen ze graag veranderd zien. De oplossing is een geautomatiseerd testsysteem. Dit testsysteem zal de websites veilig en snel houden, het zal tijd en geld besparen en het systeem zal voor kwaliteitsverhoging zorgen, omdat de websites en functionaliteiten niet meer handmatig getest hoeven te worden.

3. Doelstelling van de afstudeeropdracht

De opdracht omvat het ontwerpen en ontwikkelen van een testsysteem als onderdeel van een framework. Dit testsysteem zal tools bevatten die een deel van de testfunctionaliteiten kunnen uitvoeren. Hiervoor zal een toolselectie uitgevoerd worden. Indien er functies zijn die niet door de tools gedekt worden, zullen custom testen worden geschreven die de functionaliteiten van het framework testen. Na het uitvoeren van de opdracht zal het testen betrouwbaarder worden, doordat de testen geautomatiseerd zullen plaatsvinden en de functionaliteit van iedere website op gelijke wijze wordt getest.

4. Resultaat

Het eindproduct van de opdracht is een PoC van het testsysteem geschreven in voornamelijk PHP. Indien de geselecteerde tools hierom vragen, zal er mogelijk ook gebruik worden gemaakt van andere talen. Versies van het testsysteem worden beheerd in GIT. Het systeem zal zo gebouwd worden dat het rekening houdt met toekomstige wijzigingen, uitbreidbaarheid, testbaarheid en hergebruik. Het bedrijf zal met ingang van het geautomatiseerde testsysteem betrouwbaarder kunnen testen. De automatische tests zorgen voor producten met een hogere kwaliteit, omdat er een teststandaard gerealiseerd zal worden. Het testsysteem zal geïntegreerd worden in het bestaande framework.

5. Uit te voeren werkzaamheden, inclusief een globale fasering, mijlpalen en bijbehorende activiteiten

- Plan van aanpak schrijven (3 dagen)
- Inwerken in het framework (5 dagen)
- Bepalen welke functies er getest moeten worden en deze functies prioriteren (7 dagen)
- Het uitvoeren van een toolselectie (18 dagen)
 - o Een selectie van tools maken die de benodigde tests kunnen uitvoeren in een longlist.
 - o Onderzoek naar geselecteerde tools uitvoeren en een shortlist opstellen.
 - o Onderzoek naar de tools uit de shortlist uitvoeren.
 - o Bepalen welke tool(s) het meest geschikt zijn voor het testsysteem n.a.v. eerder bepaalde geprioriteerde te testen functionaliteit
 - o Opstellen rapport toolselectie.

- Ontwerpen van het testsysteem en de integratie van het testsysteem in het bestaande systeem (4 dagen).
- Bouwen van het systeem inclusief het integreren van de tools (16 dagen).
- Indien nodig: schrijven van custom testen (10 dagen).
- Testen van het testsysteem (7 dagen).

6. Op te leveren (tussen)producten

- Plan van aanpak
- Een lijst met de te testen functionaliteiten
- Een toolselectierapport wat omvat:
 - o Geselecteerde tools met een beschrijving van de functionaliteiten
 - o Een longlist en shortlist met geselecteerde tools
 - o Een selectie van tools die uiteindelijk gebruikt zullen gaan worden
 - o Een ontwerp van het testsysteem en de integratie van het testsysteem in het bestaande systeem
- Het testsysteem (geïntegreerd in het bestaande systeem)

7. Te demonstreren competenties en wijze waarop

- 1.3 Selecteren van standaardsoftware (niveau 3)

Het selecteren van tools die bepaalde functionaliteit van het framework kunnen testen. Het resultaat van deze beroepstaak is een onderzoeksrapport over de selectie van deze tools met selectiecriteria, long list en shortlist.

- 3.2 Ontwerpen systeemdeel (niveau 4)

Het in kaart brengen van het te bouwen testsysteem en de onderliggende structuur. Zo zal er o.a. worden bepaald hoe het testsysteem in het bestaande framework zal worden geïntegreerd. Hierbij zal gebruik worden gemaakt van UML en design patterns.

- 3.3 Bouwen applicatie (niveau 3)

Het bouwen van het testsysteem zal object georiënteerd gebouwd worden in PHP. Daarnaast zal er versiebeheer plaatsvinden via GIT. Het testsysteem zal geïntegreerd worden in het bestaande framework en er zal tijdens het bouwen rekening worden gehouden met toekomstige wijzigingen, uitbreidbaarheid, testbaarheid en hergebruik.

- 3.5 Uitvoeren van en rapporteren over het testproces (niveau 3)

Voor deze beroepstaak zal het testsysteem getest worden. Hierbij zullen de testgegevens worden gespecificeerd en zal de verwachte uitvoer worden vastgelegd.

Plan van aanpak

Optimalisering en automatisering van interne foutafhandeling procedures

De integratie van een module waarin fouten op websites worden afgehandeld en getraceerd naar de oorsprong

Geschreven door: E. Soek
Studentnummer: 11004010

Soort document: Afstudeerverslag
Examinator 1: A.M.J.J. Lousberg – Orbons
Examinator 2: E.M. van Doorn

Onderwijsinstelling: De Haagse Hogeschool
Faculteit: IT&Design
Opleiding: Informatica

In uitvoering voor: Floro Webdevelopment B.V.

Rotterdam, februari 2016

Versie 2.0

Inhoudsopgave

1	Inleiding	6
2	Aanleiding en context	7
2.1	Probleemstelling	7
2.2	Doelstelling	7
2.3	Opdrachtgever	7
3	Afbakening.....	8
3.1	Eindresultaat.....	8
4	Ontwikkelmethode	9
4.1	Iteraties	9
5	Planning.....	11
6	Herziening van de opdracht.....	12
6.1	Vernieuwde opdrachtsomschrijving	12
6.2	Gevolgen	12
6.3	Werkzaamheden.....	13
6.4	Planning.....	13

1 Inleiding

Dit plan van aanpak geeft een overzicht op het verloop van de opdracht in uitvoering voor Floro Webdevelopment B.V. en zorgt dat de betrokken partijen op één lijn zitten.

De opdracht betreft het ontwikkelen van een proof of concept van een module die zich richt op het geautomatiseerd testen van websites. Hoewel het te bouwen proof of concept in het afstudeerplan een 'testsysteem' wordt genoemd, is 'module' een meer toepasselijke term. De termen 'module' en 'proof of concept' zullen vanaf nu dan ook in het verdere verslag gebruikt worden.

De module wordt geïntegreerd in het zelf ontwikkelde framework van het bedrijf. De functie van de module is het testen van functionaliteit en het opsporen van fouten op websites die gebruikmaken van het framework. De functionaliteit van het opsporen en het achterhalen van fouten zal mede ontwikkelt worden met behulp van tools. Deze tools richten zich op het bieden van een framework voor het schrijven van testscripts. Deze tools worden geselecteerd door een toolselectie en worden geïntegreerd in de module. Soms zal er functionaliteit voorkomen die de module wel moet kunnen uitvoeren, maar wat niet ondersteund wordt door de tools. In dit geval worden de functies zelf geschreven als aanvulling op de tools.

Als de module fouten heeft opgespoord zullen deze gekoppeld worden aan wijzigingen in de code die deze fout waarschijnlijk hebben veroorzaakt. Deze wijzigingen worden opgehaald aan de hand van een connectie met de versiebeheer tool Gitlab.

Dit document beschrijft de aanleiding van de opdracht, de probleem –en doelstelling, de aanpak voor de ontwikkeling en geeft een planning voor de uitvoerperiode van de opdracht.

2 Aanleiding en context

In opdracht van Floro Webdevelopment B.V. wordt er een proof of concept ontwikkelt van een module die geautomatiseerd websites zal testen met behulp van tools die een framework bieden voor het schrijven van testscripts. Daarnaast zal de module achterhaalde fouten herleiden naar de oorsprong door gebruik te maken van versiebeheer. De module zal geïntegreerd worden in het interne framework.

2.1 Probleemstelling

In de huidige situatie worden websites die gebruikmaken van het framework handmatig getest na het uitvoeren van wijzigingen in het framework. Het bedrijf telt teveel ontwikkelde websites om handmatig één voor één te kunnen doorlopen. Hierdoor wordt er bij het testen van wijzigingen een selectie van websites gemaakt die doorlopen zullen worden. Het handmatig doorlopen van de websites en het opsporen van fouten kost veel tijd. Ook is het niet erg betrouwbaar, omdat niet gegarandeerd iedere functionaliteit doorlopen zal worden. Er is een kans dat er delen over het hoofd gezien zullen worden doordat er geen standaard set of routine aan functionaliteit is die doorlopen wordt.

De selectie van websites geven ook geen goed globaal beeld van alle websites, omdat niet iedere website hetzelfde functioneert en er fouten kunnen optreden op websites die buiten de selectie vallen. Dit heeft tot gevolg dat er fouten zullen zijn die niet achterhaald worden.

2.2 Doelstelling

Door het ontwikkelen van een module die geautomatiseerd websites kan doorlopen en testen zal er veel tijd bespaard worden door het handmatig testen te vervangen. Ook zal het proces betrouwbaarder worden omdat de module alle websites die gebruik maken van het framework zal kunnen doorlopen in plaats van een selectie hiervan. Daarnaast loopt de module een vaste set aan functionaliteit door die getest zal worden zodat op iedere website hetzelfde wordt getest en er niets zal worden vergeten.

2.3 Opdrachtgever

Floro Webdevelopment B.V. is een internetbureau met circa 25 werknemers gevestigd in het centrum van Rotterdam. Het bedrijf richt zich voornamelijk op webdevelopment en cloudoplossingen. Een aantal klanten van het bedrijf zijn: Abraham Moszkowicz, Het Gelders Orkest en Ondernemerswebsites.

3 Afbakening

De opdracht betreft het ontwikkelen van een proof of concept. Het proof of concept is een module wat geïntegreerd zal worden in het zelf ontwikkelde framework. De module zal websites die gebruik maken van het framework testen wanneer er wijzigingen zijn doorgevoerd in het framework.

De module zal voor het achterhalen van fouten gebruikmaken van tools die een framework bieden voor het schrijven van testscripts. De tools zullen gekozen worden aan de hand van een toolselectie en zullen in de module worden geïntegreerd.

Wanneer er functionaliteit voorkomt die wel getest moet worden, maar niet ondersteunt wordt door de tools wordt het testscript zelf geschreven los van de tools.

De module zal een connectie hebben met de versiebeheertool Gitlab voor het koppelen van wijzigingen in de code aan de gevonden fouten.

Om te zorgen dat de module zo compleet en doeltreffend mogelijk kan worden afgerond voor het einde van de doorlooptijd van de opdracht zal er gewerkt worden met een geprioriteerde lijst van requirements.

3.1 Eindresultaat

Het eindresultaat van de opdracht is een toolselectie, een requirementsrapport en een proof of concept van de module.

4 Ontwikkelmethode

Ik heb ervoor gekozen om deze opdracht iteratief en incrementeel uit te voeren volgens het agile principe. Omdat de opdracht niet tot in de details is uitgewerkt voor de start van de afstudeerperiode moet er ruimte zijn voor wijzigingen en aanvullingen. Niet alle requirements zijn aan het begin van de opdracht bekend. Er zullen gedurende het proces requirements bijkomen waardoor het project wendbaar moet zijn.

De volgende werkzaamheden zal ik aan het begin van de opdracht uitvoeren en vallen niet onder de iteraties, omdat deze maar eenmalig uitgevoerd hoeven te worden.

Toolselectie

Voor de opdracht zal ik een toolselectie uitvoeren voor het bepalen van de tools die geïntegreerd zullen worden in het proof of concept. De tools zullen een framework bieden voor het schrijven van testscripts. Met deze testscripts zullen er fouten op de websites die gebruik maken van het framework achterhaald worden.

Tijdens het maken van de toolselectie zal ik een longlist en een shortlist opstellen en zal ik een pilot uitvoeren waarin de tools zullen worden getest op het gebruik, de compatibiliteit en functionaliteit. Het eindresultaat van de selectie is een of meerdere tools die geschikt zijn om te integreren in het proof of concept. Deze stap wordt eenmalig uitgevoerd en zal verder niet meer geïtereerd worden, omdat dit niet benodigd is.

Inwerken

Ik zal inwerken in de interne systemen om zo kennis te vergaren en ervaring op te doen met de diversen systemen en methodes waar het bedrijf mee werkt. De systemen waarin ingewerkt zal worden zijn het Customer Relationship Management systeem (CRM), het Content Management Systeem (CMS) en het framework. Deze systemen zullen later worden toegelicht. Tijdens het inwerken zal ik gebruik maken van de documentatie van het bedrijf. Deze documentatie geeft uitleg over de diversen aspecten wat betreft de ontwikkeling van websites volgens de werkwijze van het bedrijf.

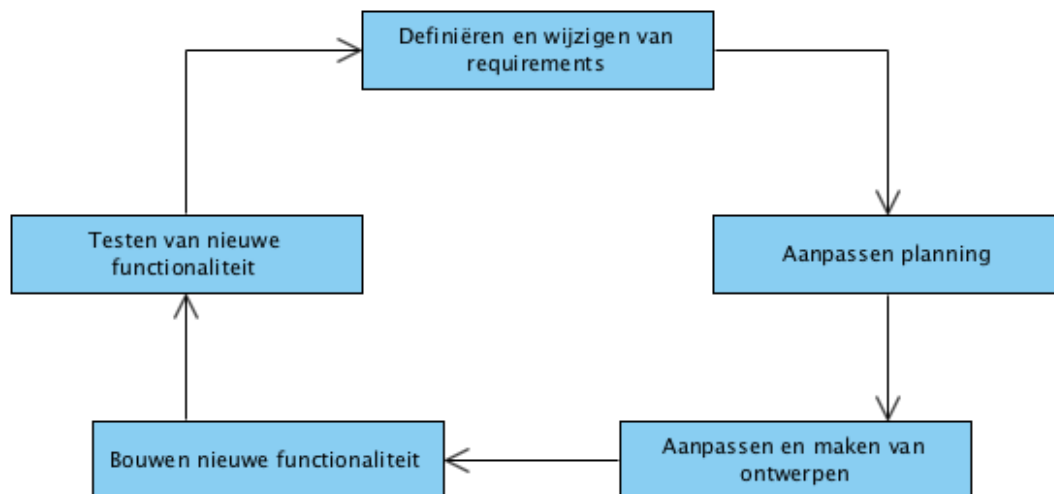
4.1 Iteraties

Gedurende de opdracht vinden er iteraties plaats gebaseerd op timeboxing en functionaliteit. De duur van de iteraties wordt bepaald door timeboxing. Voor iedere iteratie hou ik een timebox van twee weken aan. Op deze manier kan het proof of concept worden opgeleverd voor het einde van de afstudeerperiode. Iedere iteratie wordt er een nieuwe functionaliteit toegevoegd. Zo wordt het proof of concept incrementeel tot stand gebracht na iedere iteratie.

Er kunnen op dit moment nog geen iteraties worden vastgelegd, omdat er nog geen requirements bekend zijn. De iteraties zullen later in het proces gespecificeerd worden wanneer de requirements bekend zijn en er ingewerkt is.

Er zijn een aantal stappen die zullen worden uitgevoerd tijdens de iteraties. Deze stappen worden hier nader toegelicht (zie Afbeelding 1).

Afbeelding 1: De stappen tijdens de iteraties



Definiëren van (nieuwe) requirements

Aan het begin van de opdracht zal ik requirements bepalen en prioriteren aan de hand van gesprekken met de opdrachtgever. Tijdens de iteraties zullen er requirements wijzigen en requirements bijkomen. Voor het vaststellen van de requirements zal ik het boek 'handboek requirements' (Swart, 2010) erbij houden. Deze stap komt terug tijdens iedere iteratie. De requirements zullen worden vastgelegd in een requirementsrapport (zie Bijlage C).

Planning bijstellen

Tijdens de iteraties wordt indien nodig de planning aangepast. Dit kan het geval zijn als een iteratie langer of korter zal duren dan initieel ingeschat is door bijvoorbeeld gewijzigde of toegevoegde requirements. Wanneer er geen wijziging in de planning wordt vermeldt bij de iteratie, kan er vanuit worden gegaan dat de iteratie volgens de planning is verlopen.

Ontwerpen

Gedurende de opdracht zal ik ontwerpen maken om onderdelen en functies uit te werken en in kaart te brengen voordat ik deze daadwerkelijk zal bouwen. De ontwerpen zullen gemaakt worden volgens de UML notatie, waarbij gebruik wordt gemaakt van het boek 'UML 2 and the unified process' (Jim Arlow, 2011). Tijdens de iteraties zal ik ontwerpen maken voor de nieuwe functionaliteit. De ontwerpen zullen fungeren als verheldering en lijdraad voor de functionaliteit en biedt het een visueel overzicht van de werking.

Het bouwen van nieuwe functionaliteit

Wanneer de functionaliteit gedefinieerd is in de requirements en er ontwerpen zijn gemaakt kan ik beginnen met bouwen. Tijdens iedere iteratie worden er nieuwe functionaliteiten toegevoegd aan het proof of concept.

Het testen van nieuwe functionaliteit

De laatste stap van iedere iteratie is het testen van de gebouwde functionaliteit. Voor het testen zal ik het boek 'TestGoal' (Grood, 2008) raadplegen. Ik zal tijdens het testen kijken of de functionaliteit aan de requirements voldoet en of het gebouwd is volgens het ontwerp. Voor iedere functionaliteit zal er een ander type test gewenst zijn. De gebruikte testmethoden worden nader toegelicht wanneer de functionaliteiten en iteraties bekend zijn.

5 Planning

De totale afstudeerperiode zal 17 kalenderweken in beslag nemen. Dit zijn 85 werkdagen. Er zijn 70 werkdagen ingepland om aan de opdracht te werken. De overige 15 werkdagen zullen worden besteed aan het afstudeerverslag.

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Iteratie							1		2		3		4		5		
Afstudeerverslag																	
Toolselectie																	
Inwerken																	
Plan van aanpak																	
Planning																	
Requirements																	
Ontwerpen																	
Bouwen																	
Testen																	
Voortgangsgesprek																	

5.1.1 Gemaakte afspraken

Er zullen voortgangsgesprekken plaatsvinden met mijn begeleider binnen het bedrijf, Stein Fortuin. Stein is tevens de opdrachtgever. De gesprekken zullen plaatsvinden om te zorgen dat de module aan de verwachtingen blijft voldoen. Naar aanleiding van deze gesprekken zal ook het requirementsrapport worden bijgeschaafd wanneer nodig.

6 Herziening van de opdracht

De opdracht zoals deze beschreven is in de opdrachtsomschrijving en het afstudeerplan is na vier weken genoodzaakt veranderd. De verandering is tot stand gekomen wegens omstandigheden binnen het bedrijf waardoor mijn begeleider is vertrokken en ik een andere begeleider kreeg toegewezen. De verandering van begeleider/opdrachtgever, heeft gezorgd voor een nieuwe visie voor de opdracht en botsende requirements ten opzichte van de oorspronkelijke opdracht. Dit hoofdstuk beschrijft uitgebreid de herziening van de opdracht.

6.1 Vernieuwde opdrachtsomschrijving

De vernieuwde opdracht tijdens deze afstudeerperiode is het optimaliseren en automatiseren van het bedrijfsproces waarin fouten op websites en in het CMS van deze websites worden afgehandeld. Er zal naar het proces van het afhandelen van fouten (exceptions) verwezen worden als 'exception handling'. Ik zal een proof of concept bouwen van de module die dit zal realiseren. De realisatie van de vernieuwde opdracht is anders dan de oorspronkelijke opdracht.

Veranderingen ten opzichte van de originele opdracht

- De module zal niet testen met behulp van tools en testscripts zoals origineel het plan was. De module zal exceptions opsporen op websites en deze traceerbaar maken zodat deze snel kunnen worden opgelost.
- De module heeft betrekking op het CMS, het CRM en het framework.
- Er wordt een onderdeel toegevoegd dat de exceptions inzichtelijk en beheerbaar maakt.
- Het proces van de afhandeling van exceptions zal worden verbeterd.

Probleemstelling

In de huidige situatie is de exception handling van het bedrijf niet optimaal. Exceptions worden in het bedrijf behandeld als supporttickets. Supporttickets bevatten vragen van klanten en zijn het middel van communicatie met het supportteam. De tickets met exceptions komen terecht bij werknemers die bezig zijn met support en klantcontact, maar niet per definitie developers zijn. Doordat er tickets met exceptions tussen de reguliere supporttickets staan is er verminderd overzicht op de supporttickets die wel relevant zijn voor de supportafdeling.

Daarnaast heeft het bedrijf in de huidige situatie geen mogelijkheid voor het traceren van exceptions naar de oorzaak. Wanneer er een exception optreedt moet er handmatig worden achterhaald waar de exception door veroorzaakt wordt om deze te kunnen oplossen. Verder is er geen inzicht in alle opgetreden exceptions en zijn deze ook niet beheerbaar, waardoor er geen inzicht is in het oplossen van de exceptions.

Doelstelling

Het afhandelen van exceptions kan efficiënter worden uitgevoerd door deze te scheiden van de supporttickets. Om de afhandeling van exceptions te optimaliseren zal ik een proof of concept van een module bouwen waarin ik de exceptions overzichtelijk en beheerbaar zal maken. Daarnaast zal ik een tracering tot stand brengen die de oorzaak van de exception kan herleiden naar wijzigingen in de code via versiebeheer.

6.2 Gevolgen

De herziening van de opdracht heeft ook wijzigingen tot stand gebracht in de op te leveren (tussen)producten. Deze worden hier nader toegelicht.

De toolselectie

De toolselectie is niet meer benodigd voor de vernieuwde opdracht. Hoewel de toolselectie wel voor een groot deel is uitgevoerd, worden de geselecteerde tools niet meer geïntegreerd in het proof of concept en zal de pilot niet meer worden uitgevoerd.

Requirementsrapport

Voordat bekend werd dat de opdracht herzien zou worden had ik al requirements opgesteld. Omdat de opdracht een andere draai heeft gekregen kwamen er botsende requirements. Daardoor zijn de requirements die opgesteld zijn vòòr de herziening van de opdracht in een pre herziening requirementsrapport (zie Bijlage C) vastgelegd en de nieuwe requirements in een apart requirementsrapport bijgehouden (zie Bijlage E).

Beroepstaken

Door de wijziging van de opdracht is de gekozen beroepstaak 1.3: 'Selecteren van standaardsoftware' komen te vervallen en is er een nieuwe beroepstaak voor in de plaats gekomen, namelijk beroepstaak 1.4: 'Uitvoeren analyse door definitie van requirements'. Deze keuze is gemaakt, omdat door het afvallen van de toolselectie beroepstaak 1.3 niet op niveau 3 kan worden uitgevoerd. Beroepstaak 1.4 daarentegen is door de herziening heel uitgebreid uitgevoerd, daardoor is ervoor gekozen deze op niveau 4 uit te voeren.

Nieuwe begeleider/opdrachtgever

De nieuwe opdracht heeft een nieuwe begeleider/opdrachtgever met zich meegebracht. De opdracht is overgenomen van Stein Fortuin door Stephan Dorst. Stephan zal mij begeleiden gedurende de afstudeerperiode en is tevens de nieuwe opdrachtgever.

6.3 Werkzaamheden

Er worden twee werkzaamheden toegevoegd die eenmalig uitgevoerd dienen te worden. Dit zijn een procesbeschrijving van het huidige proces en het herschrijven van de huidige exception handling.

Procesbeschrijving

Tijdens de procesbeschrijving zal ik het huidige proces wat betreft de exception afhandeling van de websites vastleggen. Er zal naar dit proces verwezen worden als 'exception handling'. Om dit proces te optimaliseren zal het huidige proces in kaart worden gebracht en zal de werking ervan moeten worden begrepen.

Herschrijven van de huidige exception handling

Omdat de huidige exception handling niet optimaal functioneert en in de nieuwe situatie anders moet werken zijn er aanpassingen nodig aan dit onderdeel. De aanpassingen omvatten een verandering van de huidige exception handling en een aantal uitbreidingen. De aanpassingen en uitbreidingen zal in tijdens deze werkzaamheid vastleggen.

6.4 Planning

Doordat de opdracht is gewijzigd, is vanzelfsprekend ook de planning gewijzigd. Onderstaand is de nieuwe planning te vinden. Er zijn geen drastische veranderingen ten opzichte van de originele planning. De werkzaamheden 'procesbeschrijving' en 'herzieningen' zijn toegevoegd.

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Iteratie								1		2		3		4			
Afstudeerverslag																	
Toolselectie																	
Inwerken																	
Procesbeschrijving																	
Herzieningen																	
Plan van aanpak																	
Planning																	
Requirements																	
Ontwerpen																	
Bouwen																	
Testen																	
Voortgangsgesprek																	

Requirementsrapport

Optimalisering en automatisering van interne foutafhandeling procedures

De integratie van een module waarin fouten op websites worden afgehandeld en getraceerd naar de oorsprong

Geschreven door: E. Soek
Studentnummer: 11004010

Soort document: Afstudeerverslag
Examinator 1: A.M.J.J. Lousberg – Orbons
Examinator 2: E.M. van Doorn

Onderwijsinstelling: De Haagse Hogeschool
Faculteit: IT&Design
Opleiding: Informatica

In uitvoering voor: Floro Webdevelopment B.V.

Rotterdam, februari 2016

Versie 1.0

Inhoudsopgave

1	Inleiding	17
2	Business requirements	18
2.1	Business rules	18
2.2	Business requirements	18
3	User requirements	19
3.1	Use cases	19
4	Systeem requirements	23
4.1	Functionele systeem requirements	23
4.2	Niet-functionele systeem requirements	24
4.3	Technische beperkingen	24

1 Inleiding

Dit document beschrijft alle requirements die zijn vastgelegd aan de hand van gesprekken met de opdrachtgever. De requirements zijn onderverdeeld in: business requirements, user requirements, functionele systeem requirements en niet-functionele systeem requirements. Daarnaast zijn er ook business rules en technische beperkingen vastgelegd.

De requirements zijn geprioriteerd volgens de MoSCoW methode. MoSCoW is een prioriteringstechniek en staat voor: Must have, Should have, Could have en Wouldn't have. Op deze manier kunnen de meest belangrijke requirements als eerst in de module verwerkt worden.

De requirements in dit rapport zijn SMART geformuleerd. Het SMART-principe is een methode voor het eenduidig opstellen van doelstellingen. SMART staat voor: Specifiek, Meetbaar, Acceptabel, Realistisch en Tijdsgebonden.

Bij het opstellen van de requirements is er rekening gehouden met herleidbaarheid. Dit is terug te vinden in de tabellen aan de hand van de kolom 'BR' wat verwijst naar de business requirements en de kolom 'UC' wat verwijst naar de use cases. Herleidbaarheid wil zeggen dat de requirements altijd zijn te herleiden naar hun bronnen.

2 Business requirements

Dit hoofdstuk beschrijft alle requirements voor de business. Dit omvat business rules en business requirements.

2.1 Business rules

Business rules definiëren of beperken een bepaald aspect van het bedrijf. De module moet zich aan deze regels houden.

ID	Beschrijving	Stakeholder
R01	Bezoekers van een intern ontwikkelde website in 99% van alle bezoeken een functionerende website zien.	Stein Fortuin

2.2 Business requirements

Business requirements zijn requirements die de bestaande bedrijfsprocessen verbeteren of het zijn nieuwe processen die de opdrachtgever met de module wil uitvoeren.

ID	Beschrijving	Stakeholder
B01	De opdrachtgever wil dat het afhandelen van fouten met ingang van de module geautomatiseerd wordt.	Stein Fortuin
B02	De opdrachtgever wil dat met ingang van de module 99% van de fouten veroorzaakt door wijzigingen in het framework afgevangen worden.	Stein Fortuin
B03	De opdrachtgever wil met ingang van de module alle websites kunnen testen, in tegenstelling tot een selectie van de websites.	Stein Fortuin
B04	De opdrachtgever wil dat het proces van foutafhandeling met ingang van de module inzichtelijk gemaakt wordt.	Stein Fortuin
B05	De opdrachtgever wil dat de module geïntegreerd wordt in één van de bestaande systemen	Stein Fortuin
B06	De opdrachtgever wil dat de code die de fout veroorzaakt achterhaald wordt d.m.v. een koppeling met versiebeheer	Stein Fortuin

3 User requirements

Dit hoofdstuk beschrijft de user requirements. Dit zijn de requirements die specificeren wat de gebruiker verwacht wat de module kan doen. De gebruiker is hierbij een werknemer van het bedrijf.

ID	Beschrijving	Stakeholder	BR	UC
U01	De gebruiker wil met zo min mogelijk handelingen de module aansturen om een test uit te voeren die fouten opspoot op de ontwikkelde websites.	Stein Fortuin	B02, B03	C01
U02	De gebruiker wil met de module op ieder gewenst moment een log kunnen aanvragen van de uitgevoerde tests.	Stein Fortuin	B04	C02
U03	De gebruiker wil tijdens het uitvoeren van de testen de voortgang van de test kunnen bekijken	Stein Fortuin	B04	C03

Prioritering

ID	Must have	Should have	Could have	Won't have
U01				
U02				
U03				

3.1 Use cases

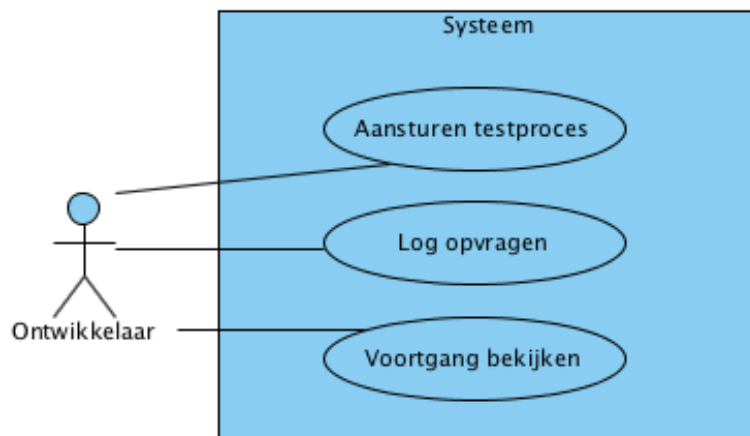
Deze paragraaf staat in het teken van use cases. Er is een use case lijst opgesteld, de use cases worden geprioriteerd, er is een use case diagram gemaakt, er is een actor lijst opgesteld en er zijn use case specificaties opgesteld.

Use case lijst

Deze lijst beschrijft de use cases die zijn afgeleid van de user requirements. De kolom 'UR' verwijst naar de bijbehorende user requirements van de use cases.

ID	Use case	Beschrijving	UR
C01	Aansturen testproces	De gebruiker wil met zo min mogelijk handelingen de module aansturen om een test uit te voeren die fouten opspoot op de ontwikkelde websites.	U01
C02	Log opvragen	De ontwikkelaar wil een log opvragen van een uitgevoerde test.	U02
C03	Voortgang bekijken	De ontwikkelaar wil de voortgang van de test kunnen bekijken tijdens het uitvoeren.	U03

Use case diagram



Actor lijst

In de volgende tabel wordt een lijst gegeven van de actors die interacteren met de module.

Actor	Beschrijving
De Ontwikkelaar	De ontwikkelaar is de persoon die een of meerdere wijzigingen heeft doorgevoerd in het framework en deze nu wil testen. De ontwikkelaar wil in een log kunnen zien of alle testen geslaagd zijn en waar het eventueel fout is gegaan.

Use case specificaties

In deze paragraaf worden er use case specificaties uitgewerkt.

Name	Aansturen testproces
ID	Spec01
Beschrijving	De ontwikkelaar wil met zo min mogelijk handelingen de module aansturen om een test uit te voeren die fouten opspoot op de ontwikkelde websites.
Primaire actors	De ontwikkelaar
Secundaire actors	-
Pre-condities	Er zijn websites aanwezig om de doorlopen
Main flow	- De ontwikkelaar geeft aan een test te willen uitvoeren door het klikken op een button in de module - De module doorloopt de ontwikkelde websites en spoort fouten op - De module geeft een overzicht van alle opgespoorde fouten
Post-condities	De websites zijn doorlopen en fouten zijn opgespoord.
Alternatieve flow	- [AF01] er zijn geen fouten gevonden

Name	Er zijn geen fouten gevonden
ID	AF01
Beschrijving	De ontwikkelaar wil met zo min mogelijk handelingen de module aansturen om een test uit te voeren die fouten opspoot op de ontwikkelde websites. Er zijn echter geen fouten gevonden.
Primaire actors	De ontwikkelaar
Secundaire actors	-
Pre-condities	Er zijn geen fouten gedetecteerd op de websites
Alternative flow	- De ontwikkelaar geeft aan een test te willen uitvoeren door het klikken op een button in de module - De module doorloopt de ontwikkelde website - De module heeft geen fouten gevonden - De module toont een melding en stelt de ontwikkelaar op de hoogte
Post-condities	De ontwikkelaar is op de hoogte gebracht dat er geen fouten gevonden zijn.

Name	Log opvragen
ID	Spec02
Beschrijving	De ontwikkelaar wil een log opvragen van een uitgevoerde test.
Primaire actors	De ontwikkelaar
Secundaire actors	-
Pre-condities	De test is uitgevoerd.
Main flow	- De ontwikkelaar navigeert naar de logsectie in de module - De ontwikkelaar geeft aan van welke test hij/zij de log wilt bekijken - De module toont de log van de gekozen test
Post-condities	Er wordt een log getoond die de uitslag van de uitgevoerde test weergeeft.
Alternatieve flow	-

Name	Voortgang bekijken
ID	Spec04
Beschrijving	De ontwikkelaar wil de voortgang van de test kunnen bekijken tijdens het uitvoeren.

Primaire actors	De ontwikkelaar
Secundaire actors	-
Pre-condities	Er wordt op het moment van het bekijken van de voortgang een test uitgevoerd.
Main flow	- De ontwikkelaar voert een test uit - De module toont de voortgang van de test
Post-condities	De module toont de voortgang van de test.
Alternatieve flow	-

4 Systeem requirements

In dit hoofdstuk komen de systeem requirements aan bod. Systeem requirements zijn de eisen die aan de module gesteld worden. Systeem requirements zijn onder te verdelen in functionele en niet-functionele systeem requirements.

4.1 Functionele systeem requirements

Functionele requirements zijn eisen die gesteld worden aan het gedrag van de module.

ID	Beschrijving	Stakeholder	BR	UC
F01	De functionaliteit in de module wordt zoveel mogelijk geschreven met behulp van de geïntegreerde tools.	Stein Fortuin	B01, B02	C01
F02	De functionaliteiten die niet met behulp van een tool geschreven kan worden, maar wel benodigd zijn worden losstaand van de tools zelf geschreven.	Stein Fortuin	B02	C01
F03	De module moet als onderdeel van het intern ontwikkelde framework geïntegreerd worden.	Stein Fortuin	B05	-
F04	De module moet de mogelijkheid bieden om door websites te kunnen navigeren.	Stein Fortuin	B01, B02	C01
F05	De module moet tijdens het uitvoeren van tests iedere door het bedrijf ontwikkelde website doorlopen.	Stein Fortuin	B01, B02, B03	C01
F06	De module moet tijdens het uitvoeren van een test een log bijhouden.	Stein Fortuin	B04	C02
F07	De module moet de mogelijkheid bieden om formulieren op websites te testen.	Stein Fortuin	B01, B02	C01
F08	De module moet de mogelijkheid bieden om links op websites te kunnen volgen.	Stein Fortuin	B01, B02	C01
F09	De module toont tijdens het uitvoeren van een test de voortgang	Stein Fortuin	B04	C03

Prioritering

ID	Must have	Should have	Could have	Won't have
F01				
F02				
F03				
F04				
F05				

F06				
F07				
F08				
F09				

4.2 Niet-functionele systeem requirements

De niet-functionele systeem requirements zijn criteria om het functioneren van de module te beoordelen. Deze requirements zijn opgesteld aan de hand van de ISO 25010. Dit is een norm die kwaliteitskenmerken van software beschrijft.

ID	Requirement	ISO	Stakeholder	Bron	BR	UC
N01	De module moet informatie kunnen uitwisselen met diverse interne systemen.	Uitwisselbaarheid - Koppelbaarheid	Stein Fortuin	Gesprek	B05	-

Prioritering

ID	Must have	Should have	Could have	Won't have
N01				

4.3 Technische beperkingen

Onderstaand zijn de technische beperkingen van de module weergegeven. Deze beperkingen geven de mogelijkheden en onmogelijkheden van de module aan.

ID	Technische beperking
T01	De module moet als onderdeel van de interne systemen geïntegreerd worden wegens beschikbare functionaliteit.
T02	De module moet PHP als hoofdtaal hebben.

Toolselectie

Optimalisering en automatisering van interne foutafhandeling procedures

De integratie van een module waarin fouten op websites worden afgehandeld en getraceerd naar de oorsprong

Geschreven door: E. Soek
Studentnummer: 11004010

Soort document: Afstudeerverslag
Examinator 1: A.M.J.J. Lousberg – Orbons
Examinator 2: E.M. van Doorn

Onderwijsinstelling: De Haagse Hogeschool
Faculteit: IT&Design
Opleiding: Informatica

In uitvoering voor: Floro Webdevelopment B.V.

Rotterdam, februari 2016

Versie 1.0

Inhoudsopgave

1	Inleiding	27
2	Criteria	28
3	Longlist	29
3.1	Criteria	29
3.2	Search	29
3.3	De lijst	29
4	Shortlist	30
4.1	JavaScript	30
4.2	PHP	30
5	Onderzoek	32
5.1	JavaScript tools	32
5.2	PHP tools	32
6	Pilot	34

1 Inleiding

Dit document beschrijft de toolselectie die is uitgevoerd voor het te bouwen proof of concept. Dit proof of concept betreft een module. Deze module zal websites die gebruikmaken van het intern ontwikkelde framework testen wanneer er wijzigingen zijn aangebracht aan het framework.

De functionaliteit van het opsporen en het achterhalen van fouten zal mede ontwikkeld worden met behulp van tools die zich richten op het schrijven van tests. Deze tools bieden een framework voor het schrijven van de testscripts en zullen geïntegreerd worden in de module.

Om de juiste tools voor de module te kiezen wordt deze selectie uitgevoerd die bestaat uit het opstellen van een longlist en shortlist. Daarnaast wordt er een pilot uitgevoerd waarin een testrun wordt uitgevoerd van de tools uit de shortlist.

2 Criteria

Voorafgaand aan de toolselectie zijn er aan de hand van gesprekken criteria achterhaald. Onderstaand zijn de criteria genoteerd die van toepassing zijn op de toolselectie.

ID	Beschrijving
C01	De tool die geïntegreerd wordt in de module moet de optie bieden om zelf testscripts te schrijven.
C02	De tool die geïntegreerd wordt in de module moet gratis beschikbaar zijn.
C03	De tool die geïntegreerd wordt in de module moet open-source zijn.
C04	De tool die geïntegreerd wordt in de module moet documentatie aanbieden.
C05	De tool die geïntegreerd wordt in de module moet een losstaande applicatie zijn om een geheel met de module te kunnen vormen.
C06	De tool die geïntegreerd wordt in de module heeft geen GUI om zo feilloos te kunnen integreren in de module.
C07	De tool die geïntegreerd wordt in de module moet JavaScript of PHP ondersteunen.
C08	De tool die geïntegreerd wordt in de module moet de websites en de content hiervan kunnen doorlopen om fouten op te sporen.
C09	De tool die geïntegreerd wordt in de module moet up-to-date zijn. De laatste update is niet langer dan een jaar geleden.

3 Longlist

De eerste stap van de toolselectie is het opstellen van een longlist. Deze longlist bevat een uitgebreide verzameling van tools die voldoen aan de opgestelde criteria. Er wordt onderscheidt gemaakt tussen tools die JavaScript ondersteunen en tools die PHP ondersteunen. Dit onderscheidt is gemaakt omdat dit twee verschillende type tools zijn die beide benodigd zijn om zowel de server-side als de client-side van de websites te kunnen testen.

3.1 Criteria

De criteria C01 en C07 zijn in acht genomen bij het opstellen van de longlist. De keuze voor deze criteria heb ik gemaakt, omdat dit de meest algemene en belangrijke criteria zijn waaraan de tools moeten voldoen. Wanneer er geen testscripts geschreven kunnen worden met de tool heeft deze geen functie binnen de module, hetzelfde geldt voor tools die geen JavaScript of PHP ondersteunen. Deze ondersteuning is nodig om de websites, die ook geschreven zijn in JavaScript en PHP, te kunnen testen.

3.2 Search

Door verschillende bronnen via Google te raadplegen zal ik een eerste selectie van tools maken voor de longlist. De termen waarop ik zoek op Google zijn: 'web test tools', 'web testing tools' en 'test framework'. Voor de longlist wil ik vooral zoveel mogelijk opties verzamelen. Deze verzameling heeft geen maximale lengte. Er zullen zowel JavaScript als PHP tools geïntegreerd moeten worden om de websites zo volledig mogelijk te kunnen testen.

3.3 De lijst

De volgende longlist(s) heb ik opgesteld n.a.v. de criteria C01 en C07. Deze tools bieden een framework en/of gelegenheid voor het schrijven van testscripts. De lijst is gescheiden in tools die JavaScript ondersteunen en tools die PHP ondersteunen.

JavaScript

- Jasmine <http://jasmine.github.io>
- CasperJS <http://casperjs.org>
- PhantomJS <http://phantomjs.org>
- SlimerJS <https://slimerjs.org>
- Mocha <http://mochajs.org>
- JSUnit <http://www.jsunit.net>
- YUI Test <http://yuilibrary.com/yui/docs/test/>
- NodeUnit <https://github.com/caolan/nodeunit>
- Tyrtle <https://github.com/spadgos/tyrtle>

PHP

- PHPUnit <https://phpunit.de/index.html>
- SimpleTest <http://www.simpletest.org>
- Snaptest <https://github.com/Jakobo/snaptest>
- Kahlan <https://github.com/crysalead/kahlan>
- PHPInlineTest <https://github.com/ptrofimov/phpinlinetest>
- Atoum <https://github.com/atoum/atoum>
- Enhance PHP <https://github.com/Enhance-PHP/Enhance-PHP>

4 Shortlist

Na het opstellen van de longlist, wordt er verder gespecificeerd naar een shortlist. Bij deze shortlist wordt er rekening gehouden met de overige criteria. Om te bepalen welke tools aan welke criteria voldoen zijn er afwegingstabellen gemaakt. Deze tabellen wegen de tools uit de longlist af tegen de opgestelde criteria.

4.1 JavaScript

Onderstaand zijn de JavaScript tools onderworpen aan de criteria. Ik heb er voor gekozen om C08 niet als knock-out criterium te beschouwen. Deze keuze is gemaakt omdat tools elkaar kunnen aanvullen in functionaliteit. Hoewel sommige tools geen websites kunnen doorlopen, hebben ze mogelijk wel andere bruikbare testmogelijkheden.

Tool	C02	C03	C04	C05	C06	C08	C09
Jasmine							
CasperJS							
PhantomJS							
SlimerJS							
Mocha							
JSUnit							
YUI Test							
NodeUnit							
Tyrtle							

Hoewel CasperJS niet losstaand is, is er toch voor gekozen deze wel op te nemen in de shortlist. Deze keuze heb ik gemaakt, omdat de tool waarvan CasperJS afhankelijk is PhantomJS of SlimerJS is. Aangezien deze tools wel aan alle criteria voldoen is het dus geen probleem om CasperJS te integreren naast PhantomJS of SlimerJS.

JSUnit heeft een GUI. Dit is niet gewenst, omdat de tool geïntegreerd wordt in de module. De tool moet feilloos opgaan in de module en alleen functionaliteit en een framework bieden die door de module gebruikt kan worden.

Daarnaast vallen JSUnit, YUI Test en Tyrtle af, omdat deze niet up to date zijn. De laatste aanpassingen aan deze tools zijn al langer dan een jaar geleden.

Naar aanleiding van de afweging is de volgende shortlist opgesteld:

- Jasmine
- CasperJS
- PhantomJS
- SlimerJS
- Mocha
- NodeUnit

4.2 PHP

Onderstaand zijn de PHP tools zijn onderworpen aan de criteria. Criterium C08 is hierbij niet in acht genomen, omdat dit alleen van toepassing is voor de JavaScript tests.

Tool	C02	C03	C04	C05	C06	C09
PHPUnit						
SimpleTest						
SnapTest						
Kahlan						
PHPInlineTest						
Atoum						
Enhance PHP						

SimpleTest is niet geschikt gebleken. De reden hiervoor is dat de tool een GUI biedt om te kunnen testen en de tool niet up-to-date is.

Verder zijn Snaptest, PHPInlineTest en Enhance PHP afgefallen, omdat de laatste aanpassing aan deze tools al langer dan een jaar geleden zijn uitgevoerd. PHPInlineTest heeft daarnaast ook geen beschikbare documentatie, wat nog een reden geeft om deze tool niet te gebruiken.

Naar aanleiding van de afweging is de volgende shortlist opgesteld:

- PHPUnit
- Kahlan
- Atoum

5 Onderzoek

Om een selectie te kunnen maken worden de tools hier nader onderzocht op functionaliteit en compatibiliteit. De tools die geschikt blijken voor integratie worden meegenomen in de pilottest en de overige vallen alsnog af.

5.1 JavaScript tools

Jasmine

Jasmine is een BDD (Behaviour Driven Development) open source testing framework voor JavaScript. Jasmine claimt een heldere en duidelijke syntax te hebben waarmee je gemakkelijk tests kan schrijven. Jasmine is gemakkelijk te installeren door het te downloaden en direct in de bestanden van de te testen web applicatie op te nemen.

CasperJS

CasperJS is een navigation scripting en testtool geschreven in JavaScript voor gebruik in combinatie met PhantomJS of SlimerJS. CasperJS stelt je in staat navigatie scenario's op websites te schrijven.

PhantomJS

PhantomJS is een tool gebaseerd op JavaScript waarmee headless getest kan worden. Dit betekent dat er via de commandline tests geschreven kunnen worden zonder dat er een GUI aan te pas komt. De tool kan functionele tests uitvoeren met frameworks zoals Jasmine en Mocha. PhantomJS op zichzelf is geen test framework. PhantomJS stelt je in staat om web pagina's te laden en manipuleren, screen captures te maken en performance te monitoren.

SlimerJS

SlimerJS is vergelijkbaar met PhantomJS. Net als PhantomJS kan SlimerJS functionele tests uitvoeren, web pagina's laden en manipuleren, screen captures maken en performance monitoren. Het enige verschil met PhantomJS is dat SlimerJS runt op Gecko, de browser engine van Mozilla Firefox in plaats van Webkit en dat SlimerJS niet volledig headless is.

Omdat SlimerJS dusdanig vergelijkbaar is met PhantomJS, met enkele nadelen ten opzichte van PhantomJS, is besloten SlimerJS niet meer mee te nemen in de selectie.

Mocha

Mocha is een JavaScript testing framework die runt op Node.js en in de browser. Node.js is een open-source runtime omgeving voor het ontwikkelen van server-side web applicaties. Voor het gebruik van Mocha zal Node.js ook geïnstalleerd moeten worden. Omdat Node.js zo'n wijds gebruikt en bekende tool is, zie ik dit niet als een knock-out factor voor deze tool. Mocha heeft verschillende features zoals: synchroon testen, asynchroon testen en veel meer features die je in staat stellen verschillende testen uit te voeren.

NodeUnit

NodeUnit maakt net als Mocha gebruik van Node.js en de browser en stelt je in staat om asynchrone unit testen te schrijven voor Node.js en de browser. NodeUnit is te installeren via een download van Github of via npm. Npm is de standaard package manager van Node.js.

5.2 PHP tools

PHPUnit

PHPUnit is een test framework voor PHP en is een onderdeel van xUnit. xUnit is een architectuur voor unit testing frameworks. xUnit is ontstaan uit sUnit en is bekend geworden met de tool JUnit.

Om PHPUnit te kunnen gebruiken kan er een PHP Archive (PHAR) bestand worden gedownload. Dit bestand kan geïnstalleerd worden met behulp van de command line.

Kahlan

Kahlan is een Unit en BDD test framework. Deze tool maakt gebruik van een describe-it syntax en is ontwikkeld op basis van het KISS (Keep It Stupid Simple) principe. De tool van gedownload worden vanaf Github en geïnstalleerd worden via Composer of Git clone.

Atoum

Atoum is wederom een PHP testing framework. Atoum is ontwikkelt met de volgende principes: het kan snel geïmplementeerd worden, het versimpelt test development en het laat het toe om betrouwbare, leesbare en heldere unit tests te schrijven. Atoum kan net als PHPUnit met een PHAR geïnstalleerd worden, maar er kan ook gekozen worden om het via Composer te installeren of het te clonen vanaf Github.

6 Pilot

De pilot is de testrun van de tools. De pilot zal uitwijzen of de tools geschikt zijn voor integratie in de module en of deze fijn in gebruik en makkelijk te begrijpen zijn.

De tools die meegenomen in de pilot zijn:

JavaScript

- Jasmine
- CasperJS
- PhantomJS
- Mocha
- NodeUnit

PHP

- PHPUnit
- Kahlan
- Atoum

6.1 Wijziging

De pilot is vroegtijdig gestopt wegens een herziening van de afstudeeropdracht waarbij de toolselectie niet meer benodigd is. De herziening van de opdracht wordt beschreven in hoofdstuk 7 van het afstudeerverslag.

Requirementsrapport

Optimalisering en automatisering van interne foutafhandeling procedures

De integratie van een module waarin fouten op websites worden afgehandeld en getraceerd naar de oorsprong

Geschreven door: E. Soek
Studentnummer: 11004010

Soort document: Afstudeerverslag
Examinator 1: A.M.J.J. Lousberg – Orbons
Examinator 2: E.M. van Doorn

Onderwijsinstelling: De Haagse Hogeschool
Faculteit: IT&Design
Opleiding: Informatica

In uitvoering voor: Floro Webdevelopment B.V.

Rotterdam, februari 2016

Versie 1.0

Inhoudsopgave

1	Inleiding	37
2	Business requirements	38
2.1	Business rules	38
2.2	Business requirements	38
3	User requirements	39
3.1	Use cases	40
4	Systeem requirements	46
4.1	Functionele systeem requirements	46
4.2	Niet-functionele systeem requirements	48
4.3	Technische beperkingen	48

1 Inleiding

Dit document beschrijft alle requirements die zijn vastgelegd aan de hand van gesprekken met de opdrachtgever. De requirements zijn onderverdeeld in: business requirements, user requirements, functionele systeem requirements en niet-functionele systeem requirements. Daarnaast zijn er ook business rules en technische beperkingen vastgelegd.

De requirements zijn geprioriteerd volgens de MoSCoW methode. MoSCoW is een prioriteringstechniek en staat voor: Must have, Should have, Could have en Wouldn't have. Op deze manier kunnen de meest belangrijke requirements als eerst in het proof of concept verwerkt worden.

De requirements in dit rapport zijn SMART geformuleerd. Het SMART-principe is een methode voor het eenduidig opstellen van doelstellingen. SMART staat voor: Specifiek, Meetbaar, Acceptabel, Realistisch en Tijdsgebonden.

Bij het opstellen van de requirements is er rekening gehouden met herleidbaarheid. Dit is terug te vinden in de tabellen aan de hand van de kolom 'BR' wat verwijst naar de business requirements en de kolom 'UC' wat verwijst naar de use cases. Herleidbaarheid wil zeggen dat de requirements altijd zijn te herleiden naar hun bronnen.

2 Business requirements

Dit hoofdstuk behandelt de business rules en business requirements.

2.1 Business rules

Business rules definiëren of beperken een bepaald aspect van het bedrijf. De module moet zich aan deze regels houden.

ID	Beschrijving	Stakeholder	Bron
R01	Bezoekers van een door het bedrijf gemaakte website dat draait op het eigen framework moeten in 99% van alle bezoeken een functionerende website zien.	Stein Fortuin	Gesprek

2.2 Business requirements

Business requirements zijn requirements die de bestaande bedrijfsprocessen verbeteren of het zijn nieuwe processen die de opdrachtgever met de module wil uitvoeren.

ID	Beschrijving	Stakeholder	Bron
B01	De opdrachtgever wil dat exception handling en tracing met ingang van de module in het CRM geautomatiseerd wordt.	Stephan Dorst	Gesprek
B02	De opdrachtgever wil dat met ingang van de module in het CRM alle exceptions op websites die draaien op het interne framework afgevangen worden.	Floris van Leerdam	Gesprek
B03	De opdrachtgever wil dat met ingang van de module in het CRM alle fouten in het CMS van websites die draaien op het interne framework afgevangen worden.	Floris van Leerdam	Gesprek
B04	De opdrachtgever wil met ingang van de module alle websites die draaien op het interne framework en het CMS van deze websites kunnen aflopen op exceptions.	Floris van Leerdam	Gesprek
B05	De opdrachtgever wil met ingang van de module in het CRM de oorzaak van de exceptions kunnen traceren.	Stephan Dorst	Gesprek
B06	De opdrachtgever wil dat er voor de module een nieuw onderdeel wordt toegevoegd aan het CRM waarin een overzicht van de exceptions wordt getoond en waarin de exceptions beheerd kunnen worden.	Floris van Leerdam	Gesprek

3 User requirements

Dit hoofdstuk beschrijft de user requirements. Dit zijn de requirements die specificeren wat de gebruiker verwacht wat de module kan doen. De gebruiker is hierbij een werknemer van het bedrijf.

ID	Beschrijving	Stakeholder	Bron	BR	UC
U01	De gebruiker wil met het onderdeel in het CRM een overzicht kunnen opvragen van de exceptions die plaatsvonden op een domein.	Floris van Leerdam	Gesprek	B06	C01
U02	De gebruiker wil met de module exceptions kunnen beheren.	Floris van Leerdam	Gesprek	B06	C02
U03	De gebruiker wil het overzicht van exceptions kunnen filteren op domein, status, prioriteit en werknemer	Floris van Leerdam	Gesprek	B06	C03
U04	De gebruiker wil de details van een exception kunnen bekijken op een detailpagina	Floris van Leerdam	Gesprek	B06	C01
U05	De gebruiker wil een status kunnen aangeven voor een exception. De status kan zijn: 'unresolved', 'being resolved' of 'resolved'.	Floris van Leerdam	Gesprek	B06	C04
U06	De gebruiker wil een prioriteit aan een exception kunnen toekennen. De prioriteit van zijn: 'high', 'medium' of 'low'.	Floris van Leerdam	Gesprek	B06	C05
U07	De gebruiker wil een werknemer aan een exception kunnen toekennen.	Floris van Leerdam	Gesprek	B06	C06
U08	De gebruiker wil een herinneringsmail kunnen versturen naar de ontwikkelaar die verantwoordelijk is voor het oplossen van een opgetreden exception.	Floris van Leerdam	Gesprek	B06	C07

Prioritering

ID	Must have	Should have	Could have	Won't have
U01				
U02				
U03				
U04				
U05				

U06				
U07				

3.1 Use cases

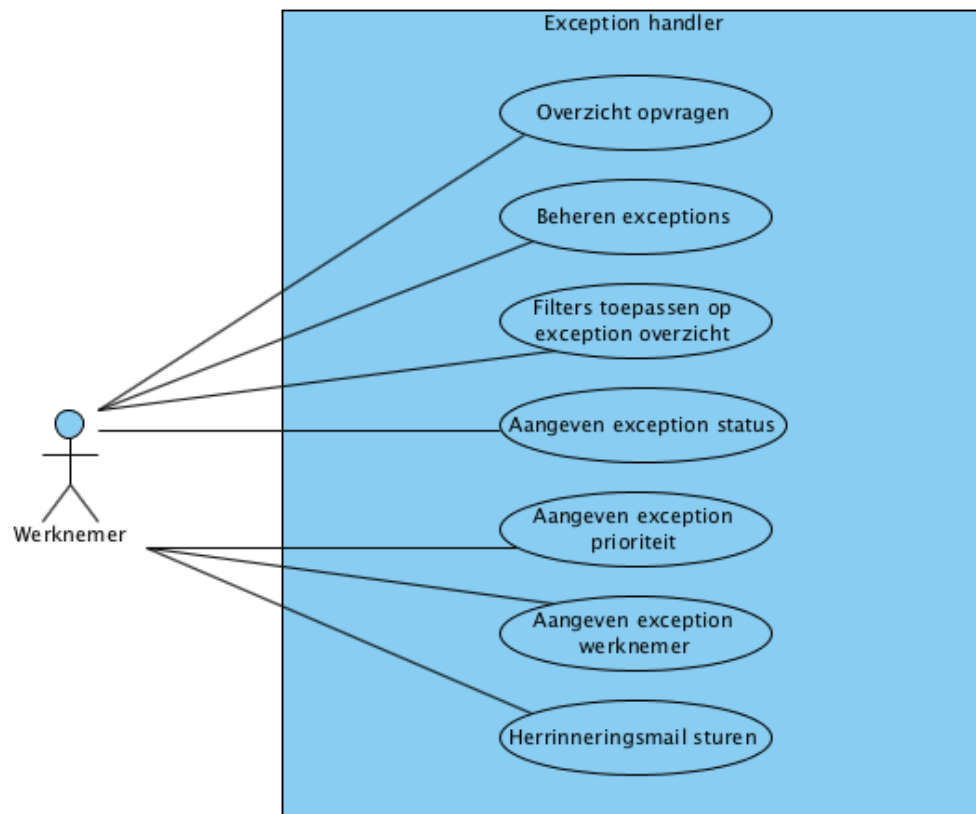
Deze paragraaf staat in het teken van use cases. Er is een use case lijst opgesteld, de use cases worden geprioriteerd, er is een use case diagram gemaakt, er is een actor lijst opgesteld en er zijn use case specificaties opgesteld.

Use case lijst

Deze lijst beschrijft de use cases die zijn afgeleid van de user requirements. De kolom 'UR' verwijst naar de bijbehorende user requirements van de use cases.

ID	Use case	Beschrijving	UR
C01	Overzicht opvragen	De werknemer wil een overzicht met exceptions kunnen ophalen van iedere website en het CMS van deze websites die draaien op het interne framework.	U01
C02	Beheren exceptions	De werknemer wil exceptions kunnen beheren.	U02
C03	Filters toepassen op exception overzicht	De werknemer wil het overzicht van exceptions kunnen filteren op domein, status, prioriteit en werknemer.	U03
C04	Aangeven exception status	De werknemer wil een status kunnen aangeven bij een exception.	U03
C05	Aangeven exception prioriteit	De werknemer wil een prioriteit aan een exception kunnen toekennen.	U05
C06	Aangeven exception werknemer	De werknemer wil een werknemer kunnen toekennen aan een exception.	U06
C07	Herrinnerings mail sturen	De werknemer wil een herrinneringsmail kunnen sturen naar de ontwikkelaar die verantwoordelijk is voor het oplossen van de exception.	U07

Use case diagram



Actor lijst

In de volgende tabel wordt een lijst gegeven van de actors die interacteren met de module.

Actor	Beschrijving
De werknemer	Een werknemer van het bedrijf wil een overzicht kunnen bekijken van de exceptions die zijn opgetreden op een bepaalde website en het CMS van deze website die draait op het interne framework. Een werknemer is niet per definitie een ontwikkelaar.
De ontwikkelaar	Een ontwikkelaar van het bedrijf kan een backend of front end developer zijn die verantwoordelijk is voor het bouwen van websites en oplossen van exceptions.

Use case specificaties

In deze paragraaf worden er use case specificaties uitgewerkt.

Name	Overzicht opvragen
ID	Spec01
Beschrijving	De werknemer wil een overzicht met exceptions kunnen ophalen van de websites en het CMS van deze websites die draaien op het interne framework.
Primaire actors	De werknemer
Secundaire actors	-

Pre-condities	Er zijn een of meerdere exceptions opgetreden op ten minste één website of het CMS van deze website dat draait op het interne framework.
Main flow	- De werknemer navigeert in het CRM menu naar het tabje 'Exceptions' onder het kopje 'Support'. - Er wordt een overzicht getoond met alle exceptions
Post-condities	Er wordt een overzicht getoond met exceptions die hebben plaatsgevonden op een website of CMS van een website die draait op het interne framework.
Alternatieve flow	- [AF01] Er is nog geen exception opgetreden

Name	Er is nog geen exception opgetreden
ID	AF01
Beschrijving	De werknemer wil een overzicht met exceptions kunnen ophalen van de websites en het CMS van deze websites die draaien op het interne framework. Er is echter nog geen exception opgetreden.
Primaire actors	De werknemer
Secundaire actors	-
Pre-condities	Er zijn nog geen exceptions opgetreden
Alternative flow	- De werknemer navigeert in het CRM menu naar het tabje 'Exceptions' onder het kopje 'Support'. - De module toont een melding dat er nog geen exceptions zijn opgetreden
Post-condities	De werknemer is op de hoogte gesteld dat er nog geen exceptions zijn opgetreden

Name	Beheren exceptions
ID	Spec02
Beschrijving	De werknemer wil de exception kunnen beheren
Primaire actors	De werknemer
Secundaire actors	-
Pre-condities	De exception is aangemaakt in de database.
Main flow	- De werknemer klikt op de link van het beheerbare attribuut van de exception die hij/zij wenst te wijzigen in het overzicht - De werknemer kiest de gewenste nieuwe waarde uit een select menu

	- De werknemer slaat de exception op
Post-condities	De exception is gewijzigd.
Alternatieve flow	-

Name	Filter toepassen op exception overzicht
ID	Spec03
Beschrijving	De werknemer wil het overzicht van exceptions kunnen filteren op domein, status, prioriteit en werknemer.
Primaire actors	De werknemer
Secundaire actors	-
Pre-condities	Er zijn exceptions aanwezig in het overzicht.
Main flow	- De werknemer kiest de gewenste filter boven het overzicht
Post-condities	Er wordt een gefilterd overzicht getoond.
Alternatieve flow	- [AF03] Een of meerdere exceptions hebben nog geen status of prioriteit

Name	Een of meerdere exceptions hebben nog geen status of prioriteit
ID	AF03
Beschrijving	De werknemer wil het overzicht van exceptions kunnen filteren op domein, status en prioriteit. Een of meerdere exceptions hebben echter nog geen domein, status, prioriteit of werknemer.
Primaire actors	De werknemer
Secundaire actors	-
Pre-condities	Er zijn exceptions aanwezig in het overzicht.
Alternative flow	- De werknemer heeft een filter geselecteerd in het overzicht - De module toont de gefilterde exceptions - De filters geven ook de optie om exceptions met een lege status, prioriteit of werknemer te tonen
Post-condities	Er wordt een gefilterd overzicht getoond.

Name	Aangeven exception status
ID	Spec04

Beschrijving	De werknemer wil een status aan een exception kunnen toekennen.
Primaire actors	De werknemer
Secundaire actors	-
Pre-condities	De exception is aangemaakt in de database.
Main flow	- De werknemer klikt op de status link van de exception die hij/zij wenst te wijzigen in het overzicht - De werknemer kiest de gewenste status uit een select menu - De werknemer slaat de nieuwe status op
Post-condities	De exception heeft een status.
Alternatieve flow	-

Name	Aangeven exception prioriteit
ID	Spec05
Beschrijving	De werknemer wil een prioriteit kunnen aangeven bij een exception.
Primaire actors	De werknemer
Secundaire actors	-
Pre-condities	De exception is aangemaakt in de database.
Main flow	- De werknemer klikt op de priority link van de exception die hij/zij wenst te wijzigen in het overzicht - De werknemer kiest de gewenste prioriteit uit een select menu - De werknemer slaat de nieuwe prioriteit op
Post-condities	De exception heeft een prioriteit.
Alternatieve flow	-

Name	Aangeven exception werknemer
ID	Spec06
Beschrijving	De werknemer wil een werknemer kunnen toekennen aan een exception.
Primaire actors	De werknemer
Secundaire actors	De werknemer

Pre-condities	- De exception is aangemaakt in de database. - De werknemer bestaat
Main flow	- De werknemer klikt op de werknemer link van de exception die hij/zij wenst te wijzigen in het overzicht - De werknemer kiest de gewenste werknemer uit een select menu - De werknemer slaat de gewijzigde exception op
Post-condities	De exception heeft een werknemer.
Alternatieve flow	-

Name	Herrinneringsmail sturen
ID	Spec07
Beschrijving	De werknemer wil een herrinneringsmail versturen naar de ontwikkelaar die verantwoordelijk is voor het oplossen van een opgetreede exception.
Primaire actors	De werknemer
Secundaire actors	De ontwikkelaar
Pre-condities	Een opgetreede exception is nog niet opgelost en er moet vaart achter gezet worden.
Main flow	- De werknemer bekijkt het overzicht van de onopgeloste exceptions - De werknemer ziet dat een belangrijke exception nog niet is opgelost - De werknemer wil de ontwikkelaar eraan herinneren deze zo spoedig mogelijk op te lossen - De werknemer klikt op een knop voor het verzenden van een herrinneringsmail voor de opgetreede exception
Post-condities	Er is een herrinneringsmail verstuurd naar de ontwikkelaar die verantwoordelijk is voor het oplossen van de opgetreede exception.
Alternatieve flow	-

4 Systeem requirements

Systeem requirements zijn de eisen die aan de module gesteld worden. Systeem requirements zijn onder te verdelen in functionele en niet functionele systeem requirements.

4.1 Functionele systeem requirements

In dit hoofdstuk komen de functionele requirements aan bod. De kolom 'BR' verwijst naar business requirements en de kolom 'UC' verwijst naar use cases.

ID	Beschrijving	Stakeholder	Bron	BR	UC
F01	De module bestaat o.a. uit een onderdeel in het CRM van het bedrijf.	Floris van Leerdam	Gesprek	-	-
F02	De module moet iedere website die op het interne framework draait en het CMS van deze websites doorlopen.	Floris van Leerdam	Gesprek	B04	-
F03	De module kan exceptions traceren naar de oorzaak in de broncode.	Floris van Leerdam	Gesprek	B05	-
F04	De module maakt exceptions beheerbaar.	Floris van Leerdam	Gesprek	B06	C02
F05	De module neemt de exceptions op in de database via een API call.	Floris van Leerdam	Gesprek	B06	-
F06	Exceptions worden opgeslagen in een aparte exceptions tabel in de database	Floris van Leerdam	Gesprek	B06	C02
F07	Het opslaan van exceptions bevat een fallback plan, waarin de exceptions gemailt worden indien deze niet via een API call kunnen worden opgeslagen.	Floris van Leerdam	Gesprek	-	-
F08	De module kan exceptions die optreden op websites die draaien op het interne framework met een Git project herleiden naar commits.	Stephan Dorst	Gesprek	B05	-
F09	De module communiceert met Gitlab via een API.	Floris van Leerdam	Gesprek	B05	-
F10	De module haalt van websites die draaien op het interne framework die een Git project hebben d.m.v. een API request naar Gitlab de laatst uitgevoerde commit op vòòr	Stephan Dorst	Gesprek	B05	-

	de optreding van een exception.				
F11	De module kan d.m.v. een API request naar Gitlab de wijzigingen van de laatst uitgevoerde commit voor de optreding van een exception ophalen	Stephan Dorst	Gesprek	B05	-
F12	Er wordt een exceptions onderdeel gebouwd in het CRM onder de categorie 'Support'.	Floris van Leerdam	Gesprek	B06	-
F13	Er wordt een overzicht getoond waarbij een filter kan worden geselecteerd voor het filteren op domein, status, prioriteit en werknemer	Floris van Leerdam	Gesprek	B06	C01, C03
F14	De module maakt het mogelijk om een status aan een exception te kunnen toewijzen.	Floris van Leerdam	Gesprek	B06	C04
F15	De module maakt het mogelijk om een prioriteit aan een exception te kunnen toewijzen.	Floris van Leerdam	Gesprek	B06	C05
F16	De module maakt het mogelijk om een werknemer aan een exception te kunnen toewijzen.	Floris van Leerdam	Gesprek	B06	C06
F17	De module maakt het mogelijk een herinneringsmail te kunnen versturen naar de developer die verantwoordelijk is voor het oplossen van een exception.	Floris van Leerdam	Gesprek	B06	C07

Prioritering

ID	Must have	Should have	Could have	Won't have
F01				
F02				
F03				
F04				
F05				
F06				
F07				
F08				

F09				
F10				
F11				
F12				
F13				
F14				
F15				
F16				
F17				
F18				

4.2 Niet-functionele systeem requirements

Dit hoofdstuk beschrijft de niet-functionele systeem requirements volgens ISO 25010. De kolom 'BR' verwijst naar business requirements en de kolom 'UC' verwijst naar use cases.

ID	Requirement	ISO	Stakeholder	Bron	BR	UC
N01	De module moet informatie kunnen uitwisselen met diverse interne systemen.	Uitwisselbaarheid - Koppelbaarheid	Floris van Leerdam	Gesprek	-	C01, C02

Prioritering

ID	Must have	Should have	Could have	Won't have
N01				

4.3 Technische beperkingen

Onderstaand zijn de technische beperkingen van de module weergegeven. Deze beperkingen geven de mogelijkheden en onmogelijkheden van de module aan.

ID	Technische beperking
T01	De module moet een onderdeel in het CRM hebben worden wegens beschikbare functionaliteit.
T02	De module moet PHP als hoofdtaal hebben.