ANALYSING SOUP

BAS BRUNINK

Een weg naar veiligere software

Software Engineering
FDMCI
Hogeschool van Amsterdam

Februari 2022 – version 0.1

[5 augustus 2021 at 16:39 – classicthesis version 0.1]

SAMENVATING

Eaglescience is een bedrijf dat innovatieve software maakt wat poogt de wereld te verbeteren. Het bedrijf is opzoek naar een methode om periodiek een automatische SOUP analyse te doen op zowel bestaande projecten als projecten die nog in ontwikkeling is. Er wordt in dit onderzoek gezocht naar die methode waarbij gekeken wordt naar een mogelijke off-the-shelf oplossing en bij het niet vinden hiervan een oplossing die in-house wordt ontwikkelt door ondergetekende.
Om inzicht te krijgen in SOUP en een analyse hierop wordt er eerst een onderzoek gedaan naar de verschillende begrippen binnen het domijn SOUP zodat er een beter beeld ontstaat in de wensen van Eaglescience.

INHOUDSOPGAVE

LIJST VAN FIGUREN

LIJST VAN TABELLEN

LISTINGS

ACRONIEMEN

API Application Programming Interface

CEO Chief Executive Officer

CFO Chief Financial Officer

COO Chief Operations Officer

CTO Chief Technology Officer

OSS Open Source Software

OTAP Ontwikkeling Test Acceptatie Productie

SOUP Software of Unkown Pedigree / Provenance

MT Management team

MoSCoW Must, Should, Could, Won't Have (zie begrippenlijst voor uitleg)

UML Unified Modeling Language

1 INLEIDING INLEIDING.TEX

Het document dat voor u ligt is een resultaat van een onderzoek en product oplevering als afstudeeropdracht door Bas Brunink voor het bedrijf Eaglescience. Het zal het process beschrijven die ik gelopen heb om een module te schrijven die automatisch een SOUP analyse doet op zowel bestaande als nieuwe projecten.

1.1 LEESWIJZER

Deze scriptie neemt de lezer mee door het verloop van het project van idee tot implementatie. Met als einde een resultaat beschrijven en een prognose in de verbetering gezien de verwachting is dat er niet direct een significante verbetering te zien is. naast een inhoudsopgave zijn er lijsten voor afbeeldingen, tabellen en listings opgenomen om snel informatie op te zoeken. Ook is er een acroniemenlijst opgenomen om de leesbaarheid voor minder ingewijden te verbeteren.

Deel I

OPDRACHT

Deel 1 gaat voornamelijk over de opdrachtgever Eaglescience en de opdracht die mij is gegeven. Hierin is te vinden wat Eaglescience voor een bedrijg is en waar het voor staat, als ook de opbouw en op welke manier er gewerkt wordt. Daarnaast is de aanvankelijke opdracht te vinden die Eaglescience heeft gegeven. Ook is te lezen hoe de opdracht vervolgens wordt ondernomen en welke aanpak er is gekozen om de opdracht tot een goed einde te brengen.

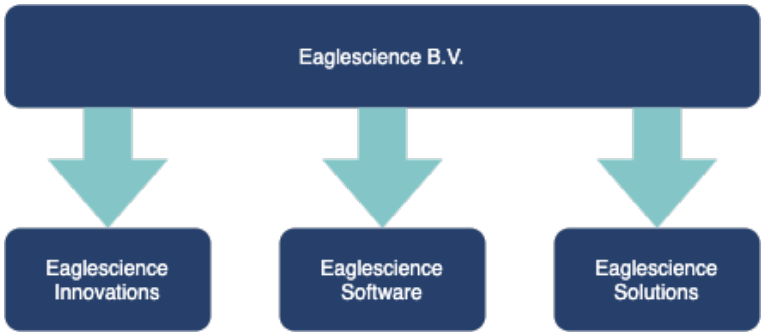
2 EAGLESCIENCE [WIP]

Het hier beschreven onderzoek en daarbij behorende applicatie is geschreven in opdracht van het bedrijf Eaglescience wat gevestigd is in Amsterdam Sloterdijk. Eaglescience ontwikkeld complexe software op projectbasis voor diverse klanten. Deze projecten hebben vaak een wetenschappelijke inslag. Diernaast biedt Eaglescience ook de mogelijkheid om zorg te dragen voor de eventuele hosting van het opgeleverde product. Eaglescience kan hierdoor nog beter garanderen dat de geboden kwaliteit in de software gewaarborgd blijft tijdens de levensduur van de software.

2.1 ORGANISATIE

Organisatie

Eaglescience bestaat uit drie divisies(Innovations, Software en Solutions) die elk onder Eaglescience BV vallen. En bestaat op het moment van schrijven uit ± 20 medewerkers waarvan 75% verantwoordelijk is voor de ontwikkeling van de geleverde software. De andere 25% bekleed een support rol zoals project managers, finance manager, quality managers, automatisering etc.



Figuur 1: Organogram Eaglescience

[betere verwoording vragen:] Eaglescience Innovations is op zoek naar nieuwe oplossingen op het gebied van software ontwikkeling en die door de software tak wordt geïmplementeert. Eaglescience Solutions is een divisie die samen met de klant opzoek gaat naar een oplossing voor een gesteld probleem. Het dagelijks bestuur is handen van:

- CEO/CFO - Marc Grootjen
- CTO - Bas Breier
- COO - Wender van Mansvelt

Onder het dagelijks bestuur valt Team Eaglescience wat bestaat uit projectmanagers en ontwikkelaars. Deze zijn onderverdeeld in diverse scrum teams die ieders verantwoordelijk zijn voor een project. De ontwikkelaars worden parallel ingezet op meerdere projecten om kennisdeling te bevorderen.

2.2 MISSIE

missie

De missie van Eaglescience is het bedienen van onze partners door een ontwerp, ontwikkeling en service te bieden op het gebied van op maat gemaakte IT oplossingen . Om dit te kunnen bewerkstelligen heeft Eaglescience goed opgeleide IT professionals in dienst die zichzelf continue ontwikkelen op de “cutting edge” van IT technologie. De hoofd competenties van de medewerkers zijn: innovatief, intelligent, klant georienteeërd, flexibel en ambitieus.

2.3 VISIE

Eaglescience streeft er als innovatief IT bedrijf naar om software te ontwikkelen als een Business-to-Business dienst. Met onze technische vaardigheden bouwen we veilige en hoogwaardige software die bijdraagt aan een betere wereld. Omdat we agile werken, leveren we precies wat nodig is, niets meer en niets minder. Wij helpen onze klanten zoeken naar een langdurige betrokkenheid en samenwerking op basis van zowel vertrouwen als wederzijds respect.

visie

Omdat elke vraag uniek is, ontwikkeld Eaglescience op maar gemaakte en innovatieve software. We zijn van plan deel uit te maken van het hele proces van het formuleren van een idee tot het lanceren van het product en het waarborgen van de productie levenscyclus. Onze belangrijkste succesfactor zijn de mensen, die zich continu ontwikkelen door met de nieuwste technieken te werken op diverse projecten. Wij streven naar een optimale balans tussen werk en privé. Dit geeft onze medewerkers veel vrijheid, maar vereist zelfdiscipline en verantwoordelijkheid.

2.4 STRATEGIE

Eaglescience levert de visie via vier strategische thema's:

- Maatschappelijke verantwoordelijkheid
- Persoonlijke groei
- Tevredenheid
- 4e????

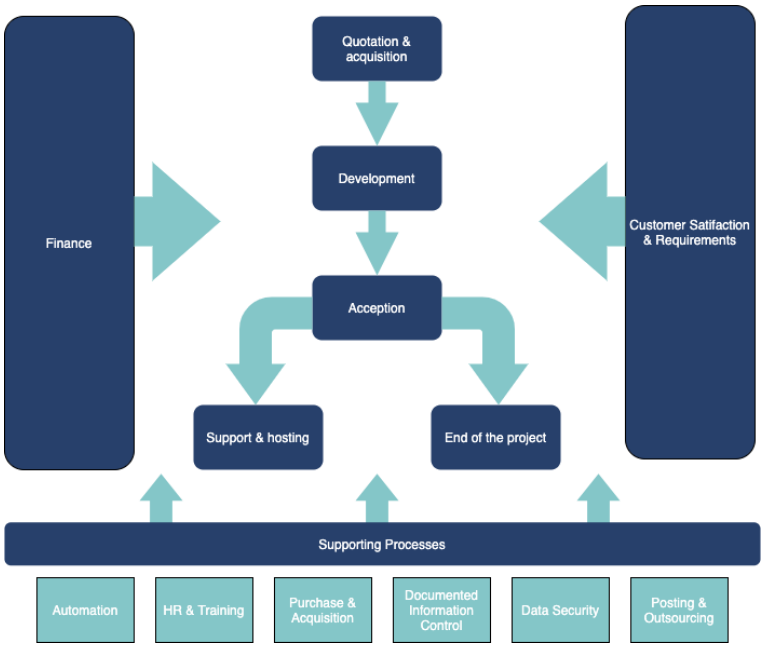
strategische thema's

We streven ernaar om veilige en hoogwaardige software diensten te leveren die waarde toevoegen aan onze samenleving. We streven naar een bedrijfscultuur waarin alle collega's hun talenten kunnen laten groeien. We hebben een ongecompliceerd werkethos: we richten ons op resultaten van hoge kwaliteit, maar met een gezonde balans tussen werk en privé en voldoende tijd voor leuke en sociale evenementen. Eaglescience verwacht van alle medewerkers dat zij hun handelen baseren op vier kwaliteitsprincipes:

- Meld situaties die niet voldoen aan onze interne procedures
- Evalueer risico's wanneer grote veranderingen worden verwacht
- Help en daag elkaar uit
- Kennis behouden over compliancy en kwaliteitsmanagement

2.5 WERKWIJZE

Zoals eerder gemeld werkt Eaglescience op projectbasis met ontwikkelaars in meedere teams. Er wordt getracht "Full scrum "te werken waarbij de requirements van de klant centraal staan. Als een project wordt aangenomen door het management team dan wordt deze in sprints in samenwerking met de klant ontwikkeld. De klant wordt nauw betrokken bij het verloop van de ontwikkeling door middel van demo's aan het einde van iedere sprint, hier wordt gemeten hoe de applicatie zich gedraagt met betrekking tot de requirements van de klant. Dit is ook het moment dat er feedback gegeven wordt en waar waar nodig gestuurd kan worden in het verdere verloop. Op het moment dat er een applicatie klaar is wordt de software dan al niet overgedragen aan de klant of door gegeven aan support en hosting die verantwoordelijk zijn voor de daadwerkelijke hosting van de software. Naast het ontwikkel process zijn er een aantal supporting processes die ervoor zorgdragen dat het bedrijf blijft draaien en er nieuwe mensen aangenomen worden, maar ook een deel automatisering die voor ondersteuning zorgt voor platformen waarop ontwikkeld en of gehosted wordt.
Eaglescience ontwikkeld op projectbasis en op die manier komen er ook inkomsten. Dus alle processen die draaien moeten ingezet kunnen worden op projecten van klanten. Als er een project voor in huis gebruik wordt ondernomen moet er een duidelijk beeld zijn of er op termijn winst mee te behalen is op monetair vlak dan al niet tijdswinst of ontwikkel gemak.



Figuur 2: Project Process

2.6

2.7 RELEVANTE EN ACTUELE ONTWIKKELINGEN BINNEN EAGLESCIENCE

Eaglescience is aan het groeien, zowel in het aantal projecten waar aan gewerkt wordt als het aantal medewerkers. Daarnaast worden de diensten die Eaglescience aanbied ook uitgebreid. Waarbij het hosten van de ontwikkelde applicaties steeds meer wordt aangeboden. Door deze inzet ligt de verantwoordelijk niet alleen bij het leveren van een veilige en hoogwaardige software maar het leveren van service waarbij de applicaties in een veilige omgeving worden aangeboden. Mede door de groei van het bedrijf maar zeker ook de diensten die aangeboden wordt is het zeer relevant om taken die geautomatiseerd kunnen worden te automatiseren.

3 OPDRACHT [WIP]

Tegenwoordig zijn software-bibliotheken niet meer weg te denken in het software ontwikkelproces van nu. Bibliotheken geven ontwikkelaars de mogelijkheid code her te gebruiken in meerdere projecten om zo efficiënter te kunnen ontwikkelen. Wat op zijn beurt weer meehelpt om de Time-To-Market te verkorten. Bibliotheken kunnen door bedrijven zelf geschreven worden, in het geval van EagseScience is dit Arches, of worden overgenomen van andere bedrijven/instellingen. Zelfs Arches is afhankelijk van een aantal bibliotheken die niet ontwikkeld zijn door Eaglescience. Dus ontkom je er tegenwoordig niet aan om bibliotheken te gebruiken waarvan je de afkomst niet geheel kan herleiden.
Deze bibliotheken vallen onder de noemer "Software of Unknown Provenance/Pedigree(SOUP)". Door het gebruik van dit soort bibliotheken kan er een aannemelijk risico ontstaan op het gebied van kwetsbaarheden. Om inzicht te krijgen in deze kwetsbaarheden en daarmee dus mogelijk veiligheidsissues dient er een SOUP analyse gedaan worden. Binnen Eaglescience wordt het belang gezien om deze analyse te doen en is daarom op zoek naar een efficiënte en mogelijk geautomatiseerde manier voor het uitvoeren van een dergelijke analyse om zo de veiligheid van de ontwikkelde applicaties te waarborgen zonder afbreuk te doen aan kwaliteit.

3.1 OPDRACHT VANUIT EAGLESCIENCE

Vanuit de CTO is de wens ontstaan om een gestructureerde methode te ontwikkelen waarbij er automatisch periodiek een SOUP analyse gedaan wordt op bestaande en nieuwe projecten. Het uiteindelijke resultaat moet zijn dat er een module wordt toegevoegd aan de reeds bestaande portal van Eaglescience waarbij project verantwoordelijken inzicht kunnen verkrijgen in de kwetsbaarheden die in een project aanwezig kunnen zijn door het gebruik van externe bibliotheken.

3.1.1 *Eisen aan de opdracht*

Vanuit Eaglescience zijn er een aantal eisen gesteld waaraan het eindproduct moet voldoen. Als er aan deze eisen is voldaan dan is er voor Eaglescience een waardevol product wat men dan ook in gebruik kan nemen. Daarnaast zijn er een aantal oplever eisen die gehaald dienen te worden om de kwaliteit te waarborgen.

functionele eisen

- De module dient eenvoudig te worden gebruikt in de huidige CI/CD pipeline voor bestaande en nieuwe projecten
- De module dient gebruik te maken van de bestaande ++huidige++ projectstructuur van het portal
- De module dient ondersteuning te bieden voor meerdere omgevingen(OTAP)
- De module dient met een instelbaar interval de analyse uit te voeren
- De module op project en omgeving niveau te rapporteren over bekende kwetsbaarheden
- De module dient kwetsbaarheden op minimaal drie niveau's in te schalen (kritisch, gemiddeld en laag)
- De module dient ondersteuning te bieden voor het instellen van quality gates ten aanzien van ieder niveau, per project, per omgeving
- De module wordt ontwikkeld in Angular en Play(scala), overeenkomstig bestaande portal modules

kwaliteitseisen

- De module voldoet aan de geldende kwaliteitsnormen binnen Eaglescience, minimaal meetbaar door:
  - test coverage > 70%
  - onderdeel van de bestaande CI/CD voor het Eaglescience Portal
- Geschreven code is gereviewd door een Eaglescience ontwikkelaar
- In de module zijn gescheiden componenten: Frontend, Backend, API onafhankelijk en goed gedocumenteerd.
- Voor de API documentatie wordt gebruik gemaakt van swagger.

3.1.2 *Deliverables*

Vanuit de CTO zijn er naast de functionele eisen ook eisen gesteld aan de oplevering:

- Geïntregreerde en aantoonbaar werkende module
- De code van de module in Eaglsescience GitLab
- API documentatie (middels swagger)
- Een handleiding hoe de module gebruikt dient te worden
- Eventuele aanvullende deliverables vanuit de HvA

3.2 OPDRACHT FASEN

Om de hierboven beschreven opdracht zo goed als mogelijk uit te voeren dient er eerst een onderzoek gedaan worden naar het begrippen binnen het domein SOUP en daarnaast mogelijkheden om bibliotheken te screenen/testen op kwetsbaarheden. Daarna moet er een module ontwikkeld worden die deze mogelijkheid implementeerd met in achtneming van de bovengenoemde eisen.

3.2.1 *Fase 1: Onderzoek*

Als eerste dient er een begrippen/literatuur onderzoek gedaan te worden binnen het domein soup om een goed kennis te vergaren over het domein om zo een basis te leggen voor een te implementeren module. Daarnaast dient er onderzoek gedaan te worden om te zien of er bibliotheken zijn en resources waar informatie of SOUP bibliotheken te vinden is. en aan welke eisen deze moeten voldoen. Hier lettende op de eisen vanuit Eaglescience en de mogelijkheden die deze analyse bibliotheken bieden. Deze fase wordt beschreven in het tweede deel van dit document.

3.2.2 *Fase 2: Oplevering SOUP analyse module*

De uit het onderzoek behaalde resultaten ten opzicht van beschikbare resources om een SOUP analyse te voeden moet worden geimplementeerd in een module binnen een bestaande applicatie. Deze module dient te voldoen aan de eisen die gesteld zijn. Het ontwerp en implementatie wordt beschreven

3.3 PLAN VAN AANPAK

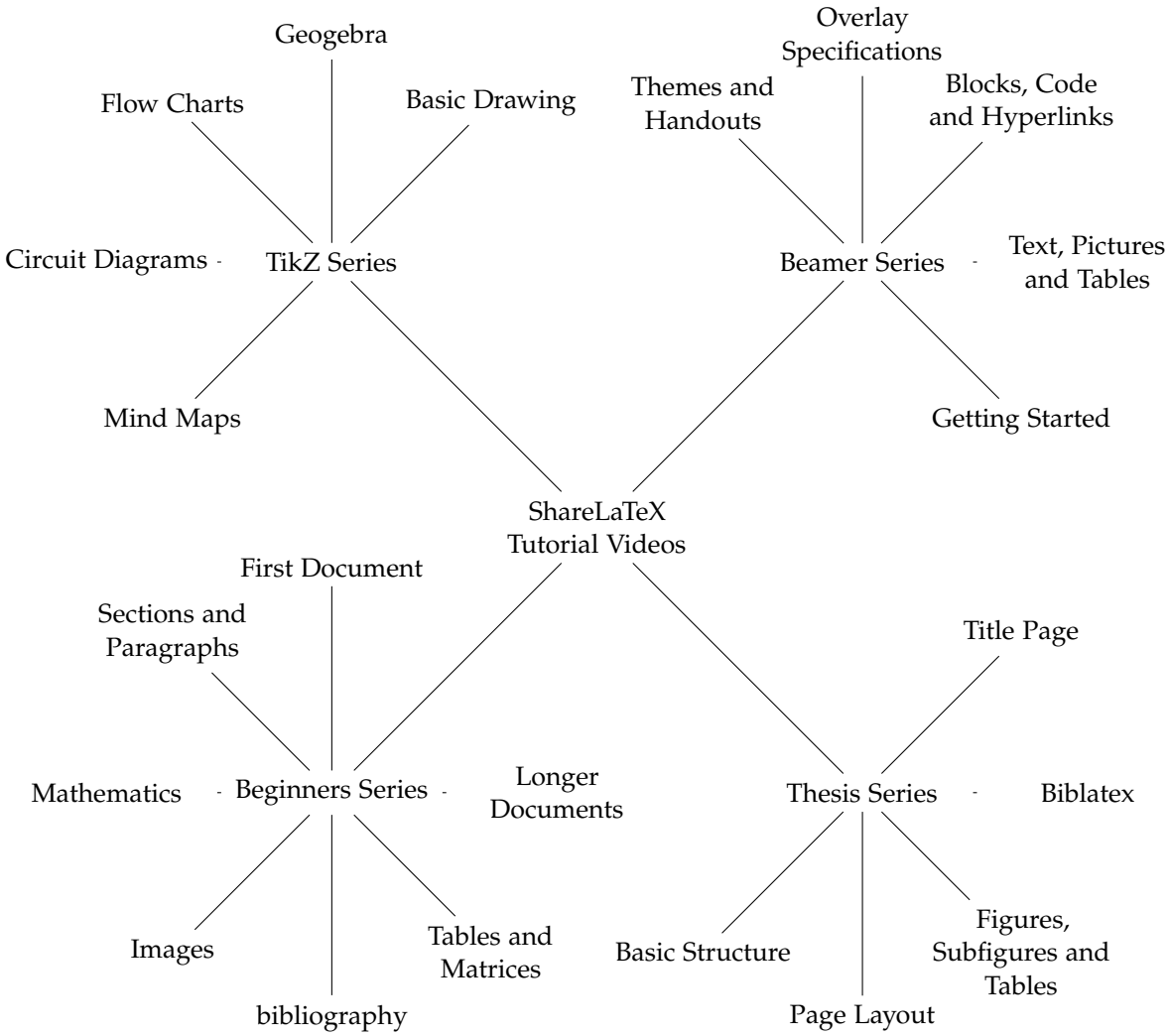
Het plan is als volgt:

1. LiteratuurOnderzoek
   Wat gebeurt er als ik hier text plaats

2. Markt onderzoek
3. Resultaat onderzoek
4. ontwerp implementatie
5. Ontwikkeling implementatie
6. Deploy implementatie

3.4 MINDMAP TEST

Kijken of dit nog waardevol is :



Deel II

REQUIREMENTS ANALYSE EN PLANNING

inleidende tekst over de Requirement analyse en planning

4 REQUIREMENTS ANALYSE

Om inzicht te krijgen in de eisen van de nieuwe module naast de eisen die al vermeld staan in de opdracht. is er een intake gesprek geweest met de CTO (Bas Breier), in dit gesprek is aan bod gekomen wie de stakeholders zijn en welke requirements hij heeft naast de requirements die in de opdracht staan. In dit gesprek is er meer ingegaan op de details van het functioneren van de module. Ook is er een beeld geschets over de huidige situatie en naar welke situatie er gegaan moet worden.

4.1 HUIDIGE SITUATIE

In de huidige situatie wordt er een SOUP analyse gedaan door de ontwikkelaars op het moment dat een project ontwikkelt wordt. dit is veelal handmatig zoeken in online resources op bibliotheken die gebruikt worden. Dit neemt veel kostbare tijd in beslag die beter besteed kan worden om nieuwe features toe te voegen. Daarnaast worden de bevindingen die gedaan worden niet centraal opgeslagen zodat er een potentie is dat niet iedereen op de hoogte is van de actuele informatie.

4.2 GEWENSTE SITUATIE

De gewenste situatie is dat projectmanagers, ontwikkelaars en het dagelijks bestuur real-time inzage hebben in de huidige staat van de kwetsbaarheden in de gebruikte externe bibliotheken. Dit is te doen door een onderdeel in een portal te bouwen die op een overzichtelijke manier deze informatie weergeeft.

om de informatie weer te kunnen geven moet er een manier worden gevonden om tijdens een bouwprocess de versies van de gebruikte bibliotheken te achterhalen. En deze vervolgens tegen een Vulnerability Database te leggen. Deze gegevens dienen in een interne database opgeslagen te worden waarop de het onderdeel in de portal gegevens op kan halen. Waarop vervolgens door de stakeholders de gewenste informatie gehaald kan worden.

4.3 DE STAKEHOLDERS

De stakeholders kunnen opgedeelt worden in twee hoofdgroepen(Tabel 1): externe en interne stakeholders. De klant is als enige een externe stakeholder en is buiten de analyse gehouden omdat deze staceholder alleen indirect resultaten bemerkt doormiddel van het verkrijgen van verbeterde software. De klant als stakeholder is dus passief en zal niet worden geinterviewt. Met personen uit de interne stakeholders groepen zijn interviews gehouden om een inzicht te krijgen in hun belang en invloed bij de module. Daarnaast is er een eerste lijst met requirements opgesteld waaraan de module dient te voldoen. Deze lijst is een start en zal na enkele sprintdemo's worden aangepast of uitgebreid. Na iedere tweede sprint zal een evaluatie worden gehouden om inzicht te krijgen of de requirements nog accuraat zijn en eventueel nog moeten worden aangescherpt. De belangen en invloeden worden in de komende subsecties verder toegelicht.

4.3.1 *Dagelijks bestuur (intern)*

Het dagelijks bestuur ziet vooral voordelen in het inzicht krijgen van kwetsbaarheden op een overzichtelijke manier, zodat ze kunnen sturen in het gebruik van biblioteken of andere technologiën. Echter zien zij ook kosten gemoeid met de verandering. Door de manier van werken dienen deze kosten te-

GROEP	STAKEHOLDER
Extern	Klant
Intern	Dagelijks Bestuur
	Project managers
	CTO
	Ontwikkelaars

Tabel 1: Verdeling stakeholders

rug verdient te worden door werkzaamheden binnen andere projecten. De CTO ziet vooral tijdswinst zodat de time-to-market voor andere projecten hoger ligt en dus meer verdient kan worden.

4.3.2 *Projectmanagers (intern)*

Project managers krijgen op dit moment een update over de staat van kwetsbaarheden tijdens stand-ups en aan het einde van een sprint tijdens de sprint demo's. De nieuwe module biedt ze de mogelijkheid om up-to-date informatie on-demand te verkrijgen. Op de vraag of het het waard is dat een aantal ontwikkelaars tijd kwijt zijn in testen en meedenken over de module weegt volgens hen op tegen de voordelen die de module in de toekomst kan brengen.

4.3.3 *Ontwikkelteam (intern)*

Het ontwikkelteam wil graag meedenken en meewerken aan een oplossing, gezien zij de gene waren die handmatig de analyse uitvoerden. Zij zien voor een oplossing voor een taak dat veel tijd in beslag nam en afleide van de daadwerkelijke taak.

4.3.4 *Klant(extern)*

Als laatste de klant welke een passieve stakeholder is gezien zij niet direct betrokken zijn bij de ontwikkeling van de module maar wel verbeteringen genieten ind e zin van veilige en betrouwbare software.

4.3.5 *Stakeholder analyse*

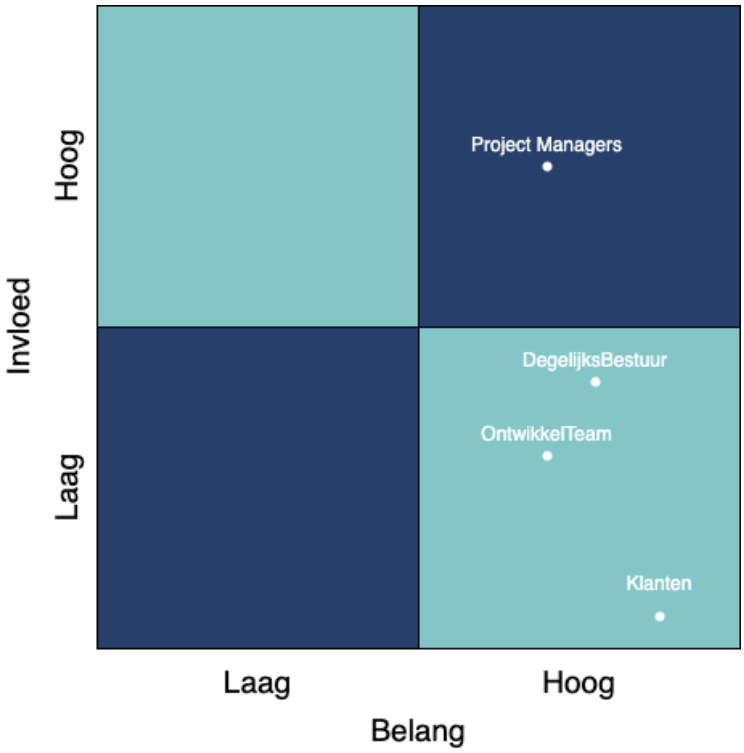
Zoals te zien is in figuur[X] zijn de projectmanager, het ontwikkelteam en de klanten het meest gebaad bij een nieuwe module voor de analyse van kwetsbaarheden. Echter zijn de klanten niet tot bijna niet betrokken bij de ontwikkeling van de module maar hebben er indirect wel belang bij omdat de software die voor hen ontwikkeld wordt veiliger wordt door het voeren van een geautomatiseerde analyse. Door deze analyse worden alleen de requirements meegnomen die intern zijn opgenomen.

4.4 REQUIREMENTS

Naast het analyseren van de betrokkenheid en belang van de stakeholders is er ook gevraagd welke requirements ze terug wilden zien in de applicatie en welke prioriteit er aan gesteld werdt. Om een leidraad te verschaffen is de MoSCoW-methode gebruikt. Hieronder is een lijst geformuleert met de belangrijkset requirements vanuit de stakeholders. Deze lijst is niet volledig en wordt na iedere sprint aangepast aan de resultaten van de sprint ervoor.

Must Have

- Als *Dagelijks Bestuur* wil ik ... zodat ik...



Figuur 3: StakeHolders Analyse

- Als *Dagelijks Bestuur* wil ik ... zodat ik...
- Als *Dagelijks Bestuur* wil ik ... zodat ik...
- Als *Dagelijks Bestuur* wil ik ... zodat ik...
- Als *Ontwikkel team* wil ik ... zodat ik...
- Als *Ontwikkel team* wil ik ... zodat ik...
- Als *Ontwikkel team* wil ik ... zodat ik...
- Als *Ontwikkel team* wil ik ... zodat ik...
- Als *Ontwikkel team* wil ik ... zodat ik...
- Als *Project manager* wil ik een overzicht per project kunnen zien met daarin de gebruikte bibliotheken zodat ik inzage heb ik wat er gebruikt wordt voor ontwikkeling.
- Als *Project manager* wil ik een overzicht per project kunnen zien met daarin de kwetsbaarheden die gevonden zijn en welke bibliotheken er geupdate moeten worden om deze kwetsbaarheden aan te kunnen pakken.
-

Should Have

- Als *Dagelijks Bestuur* wil ik ... zodat ik...
- Als *Dagelijks Bestuur* wil ik ... zodat ik...
- Als *Dagelijks Bestuur* wil ik ... zodat ik...
- Als *Dagelijks Bestuur* wil ik ... zodat ik...

- Als *Ontwikkel team* wil ik ... zodat ik...
- Als *Ontwikkel team* wil ik ... zodat ik...
- Als *Ontwikkel team* wil ik ... zodat ik...
- Als *Ontwikkel team* wil ik ... zodat ik...
- Als *Ontwikkel team* wil ik ... zodat ik...
- Als *Project manager* wil ik een overzicht per project kunnen zien met daarin de gebruikte bibliotheken zodat ik inzage heb ik wat er gebruikt wordt voor ontwikkeling.
- Als *Project manager* wil ik een overzicht per project kunnen zien met daarin de kwetsbaarheden die gevonden zijn en welke bibliotheken er geupdate moeten worden om deze kwetsbaarheden aan te kunnen pakken.
-

Could Have

- Als *Dagelijks Bestuur* wil ik ... zodat ik...
- Als *Dagelijks Bestuur* wil ik ... zodat ik...
- Als *Dagelijks Bestuur* wil ik ... zodat ik...
- Als *Dagelijks Bestuur* wil ik ... zodat ik...
- Als *Ontwikkel team* wil ik ... zodat ik...
- Als *Ontwikkel team* wil ik ... zodat ik...
- Als *Ontwikkel team* wil ik ... zodat ik...
- Als *Ontwikkel team* wil ik ... zodat ik...
- Als *Ontwikkel team* wil ik ... zodat ik...
- Als *Project manager* wil ik een overzicht per project kunnen zien met daarin de gebruikte bibliotheken zodat ik inzage heb ik wat er gebruikt wordt voor ontwikkeling.
- Als *Project manager* wil ik een overzicht per project kunnen zien met daarin de kwetsbaarheden die gevonden zijn en welke bibliotheken er geupdate moeten worden om deze kwetsbaarheden aan te kunnen pakken.
-

Won't Have

- Als *Dagelijks Bestuur* wil ik ... zodat ik...
- Als *Dagelijks Bestuur* wil ik ... zodat ik...
- Als *Dagelijks Bestuur* wil ik ... zodat ik...
- Als *Dagelijks Bestuur* wil ik ... zodat ik...
- Als *Ontwikkel team* wil ik ... zodat ik...
- Als *Ontwikkel team* wil ik ... zodat ik...
- Als *Ontwikkel team* wil ik ... zodat ik...
- Als *Ontwikkel team* wil ik ... zodat ik...

- Als *Ontwikkel team* wil ik ... zodat ik...
- Als *Project manager* wil ik een overzicht per project kunnen zien met daarin de gebruikte bibliotheken zodat ik inzage heb ik wat er gebruikt wordt voor ontwikkeling.
- Als *Project manager* wil ik een overzicht per project kunnen zien met daarin de kwetsbaarheden die gevonden zijn en welke bibliotheken er geupdate moeten worden om deze kwetsbaarheden aan te kunnen pakken.
-

De Won't haves staan hierbij genoemd als leidraad voor eventueel updates in de toekomst. Als blijkt dat er tussen de won'ts toch low hanging fruit blijkt te hangen kunnen deze meegenomen worden in de sprints. De requirements worden als epics in een JIRA omgeving gezet om vervolgens een planning te kunnen maken.

4.5 WERKWIJZE EN PLANNING

Binnen Eaglescience wordt er scrum gewerkt en ook al ben ik als enig werkzaam op dit project zal er zoveel mogelijk op deze manier worden gewerkt inhoudend dat een sprint 2 weken duurt met aan het begin een springplanning en aan het einde van de sprint een demo en een retrospective zal worden gehouden. De daily stand-ups zal worden gehouden met de Product-owner en de reviewers van de code om zo een kortere feedback loop te krijgen. Daarnaast staan er verschillende collega's die support kunnen leveren.

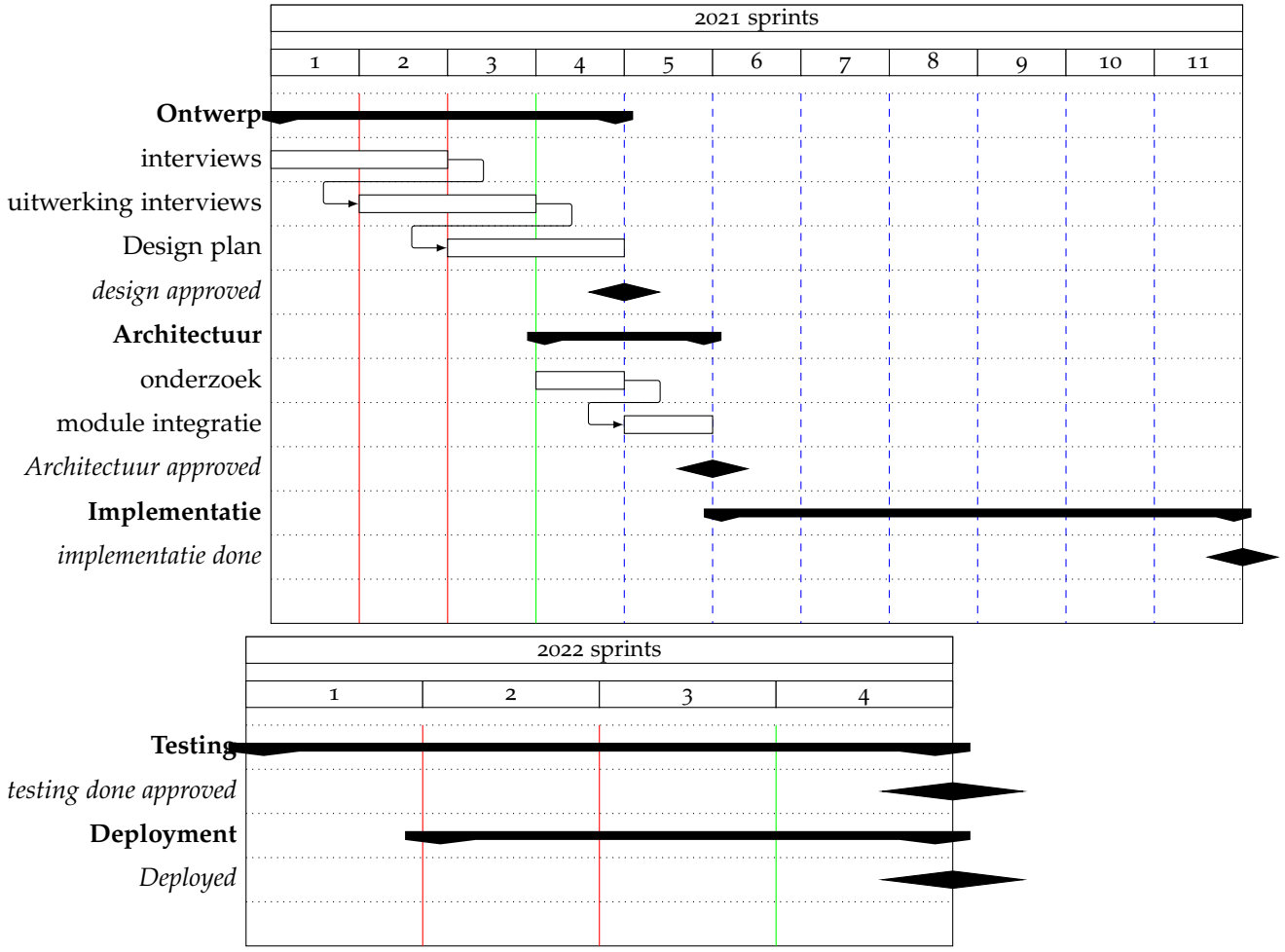
5 PLANNING

5.1 PLANNING METHODE

Binnen Eaglescience wordt er gewerkt middels de Agile Scrum methode, wat inhoud dat elk project incrementeel opgeleverd wordt in sprints. De Gant Chart hieronder is dan ook ingedeeld in sprints van twee weken en geeft alleen de hoofd werkzaamheden weer. De gedaileerde planning zal worden gedaan middels een Scrum board in Jira. Welke iedere sprint zal worden gereviewed. en aangepast aan de daadwerkelijke stand in het project.

5.2 PROJECT PLANNIN IN GROTE LIJNEN.

test



Figuur 4: Planning

Deel III

ONDERZOEK

text deel 2

6 INLEIDING

Op basis van de requirements analyse beschreven in het vorige deel zijn er een aantal vragen ontstaan die verder onderzoek benodigd behoeven. In dit deel worden de vragen geanalyseerd en beantwoord zodat er een duidelijkheid is in de materie en een goede basis wordt gelegd voor de daadwerkelijke implementatie beschreven in het volgende deel.

6.1 SCOPE

Het onderzoek zal zich beperken tot de benodigde informatie voor het implementeren van de nieuwe oplossing voor een geautomatiseerde SOUP analyse. Het zal ingaan op de gebruikte ontwikkelstack binnen Eaglescience en bestaande architectuur gezien de nieuwe oplossing een onderdeel is van een al bestaand project en hier dus naatloos op moet integreren. Aan de orde zullen komen:

- Wat is een SOUP analyse?
  - Welke oplossingen bestaan er op dit moment om een SOUP analyse te doen?
  - Is een API / Database waar kwetsbaarheden in opgesomt zijn?
- Wat is de ontwikkel stack waar Eaglescience mee werkt?
- Hoe ziet de portal er op dit moment uit en hoe is het nieuwe onderdeel hierin te integreren?

In feite zijn er twee onderzoeken die gedaan moeten worden. ten eerste is er een onderzoek naar de gebruikte middelen binnen eaglescience en daarnaast een theoretisch onderzoek naar termen binnen SOUP analyse.

In het komende hoofdstuk wordt de methode duidelijk hoe de onderzoeken gedaan worden. en de hoofdstukken daarna zullen de vragen en de daarbij horende antwoorden beschrijven.

7 ONDERZOEKSMETHODE

Zoals in de inleiding vermeld zijn er twee onderzoek benodigd om deze opdracht tot een goed einde te brengen. in dit hoofdstuk wordt uitwijd over de methoden die zijn toegepast om antwoorden te verkrijgen op de vragen.

7.1 ONDERZOEKSMETHODE ARCHITECTUUR BINNEN EAGLESCIENCE

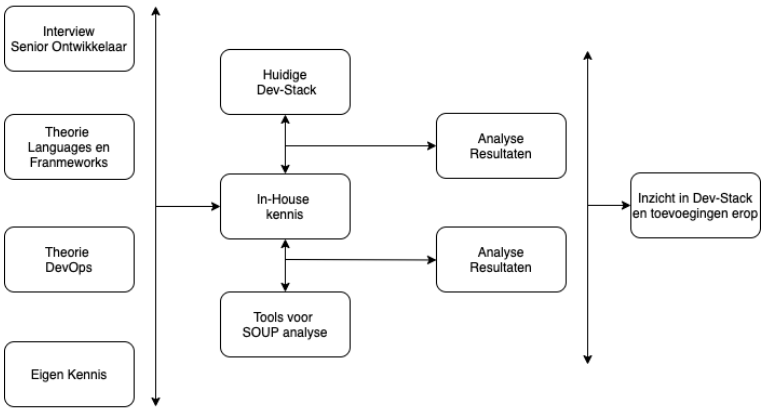
7.1.1 *Doel van het onderzoek*

Het doel van het onderzoek is het in kaart brengen van de dev-stack die bij Eaglescience gebruikt wordt. Deze kennis is noodzakelijk om een goede architectuur te kunnen opzetten die past binnen de huidige manier van werken maar toch de faciliteiten biedt die nodig zijn om de nieuwe oplossing bruikbaar te maken voor alle stakeholders. Daarnaast is door het in kaart brengen van de huidige dev-stack ook de mogelijkheid om te onderzoeken of er toevoegingen zijn die het implementeren van deze oplossing kunnen vergemakkelijken.

Om dit doel te bereiken is er als eerst een interview geweest met een senior ontwikkelaar op basis van dit interview is verder onderzocht welke bibliotheken en tools er beschikbaar zouden zijn om de nieuwe oplossing te ontwikkelen.

7.1.2 *Onderzoeksmodel*

Het onderstaande onderzoeksmodel is samengesteld uit de vragen aan de linkerkant en de weg naar het resultaat van links naar rechts.



Figuur 5: Onderzoeksmodel Eaglescience

7.1.3 *Onderzoeks vragen*

De hoofdvraag in dit onderzoek luid:
"Waaruit bestaat de huidige Dev-stack en welke tooling missen we om een geautomatiseerde SOUP analyse te doen?"
Uit deze hoofdvraag komen een aantal deelvragen:

- Hoe wordt op dit moment gewerkt binnen Eaglescience en dan met name op het gebied van SOUP analyses?
- Welke Ontwikkeltalen gebruiken we binnen Eaglescience?
- Welke frameworks worden er gebruikt binnen de ontwikkeltalen?
- Hoe wordt op dit moment de ontwikkelde software gedeployed?
- Welke architectuur wordt er op dit moment gebruikt in de portal?
- Waar wordt de softwar uiteindelijk gedeployed?
- Nog veel meer vast??

7.1.4 *Resultaat*

Het resultaat van dit onderzoek is het hebben van een beeld hoe software op dit moment wordt ontwikkeld binnen Eaglescience en hoe deze vervolgens wordt gedeployed en waar. Ook wordt er inzicht verschaft in de mogelijkheden om de nieuwe oplossing te implementeren gezien deze een onderdeel moet gaan worden van de bestaande deploy-pipeline.

7.1.5 *Strategie*

De beste manier om de antwoorden op deze vragen te krijgen is door het interviewen van de ontwikkelaars en de CTO. Deze hebben op dit moment de meeste kennis van de gebruikte systemen binnen Eaglescience. Ook de verschillende ärtifact files"(Package.json / build.sbt) zijn goede bronnen om te onderzoeken welke frameworks er gebruikt worden. het gaat dan voornamelijk over de build tools waar informatie uit te halen is over bibliotheken en versies hiervan.

7.2 ONDERZOEK NAAR SOUP ANALYSE

7.2.1 *Doel van het onderzoek*

Het doel van dit onderzoek is het opbouwen van een theoretische basis voor het ontwikkelen van de SOUP module. Alsmede methodes om dit geautomatiseerd te kunnen doen in combinatie met databases waar kwetsbaarheden in opgeslagen zijn.

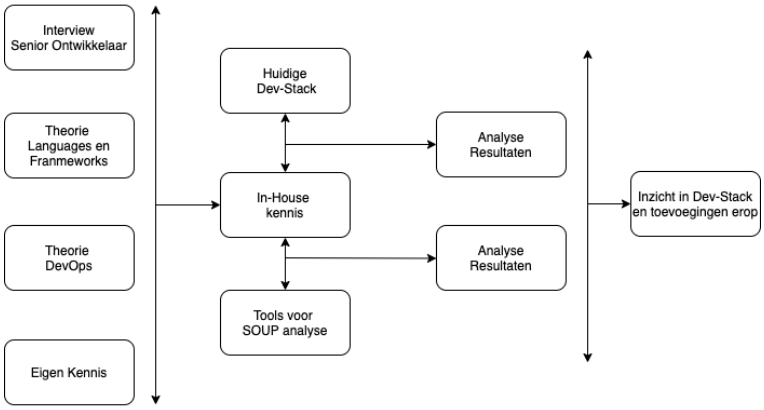
7.2.2 *Onderzoeksmodel*

Het onderstaande onderzoeksmodel is samengesteld uit de vragen aan de linkerkant en de weg naar het resultaat van links naar rechts.

7.2.3 *Onderzoeks vragen*

De hoofdvraag in dit onderzoek luid:
"Waaruit bestaat de huidige Dev-stack en welke tooling missen we om een geautomatiseerde SOUP analyse te doen?"
Uit deze hoofdvraag komen een aantal deelvragen:

- Hoe wordt op dit moment gewerkt binnen Eaglescience en dan met name op het gebied van SOUP analyses?
- Welke Ontwikkeltalen gebruiken we binnen Eaglescience?
- Welke frameworks worden er gebruikt binnen de ontwikkeltalen?



Figuur 6: Onderzoeksmodel Eaglescience

- Hoe wordt op dit moment de ontwikkelde software gedeployed?
- Welke architectuur wordt er op dit moment gebruikt in de portal?
- Waar wordt de softwar uiteindelijk gedeployed?
- Nog veel meer vast??

7.2.4 *Resultaat*

Het resultaat van dit onderzoek moet zijn dat er een theoretische basis is voor het verdere verloop van het project.

7.2.5 *Strategie*

Dit onderzoek is voor een groot deel een bureauonderzoek uit bronnen online en boeken. Waarbij er een verslag wordt gelegd die geverifieerd wordt door de opdrachtgever.. De beste manier om de antwoorden op deze vragen te krijgen is door het interviewen van de ontwikkelaars en de CTO. Deze hebben op dit moment de meeste kennis van de gebruikte systemen binnen Eaglescience. Ook de verschillende ärtifact files"(Package.json / build.sbt) zijn goede bronnen om te onderzoeken welke frameworks er gebruikt worden. het gaat dan voornamelijk over de build tools waar informatie uit te halen is over bibliotheken en versies hiervan.

7.3 TIJDSVERLOOP ONDERZOEKEN

Beide onderzoeken zullen parallel uitgevoerd worden zodat beide onderzoeken elkaar inzichten kunnen verschaffen. en er op die manier een beter begrip van de mogelijkheden is.

Deel IV

APPENDIX

Dit deel bevat extra informatie die belangrijk kunnen zijn voor het begrip van het onderzoek maar niet echt plaats hebben in het document zelf.

A

BEGRIPPENLIJST [WIP]

Time to Market
De marktintroductietijd is de tijdsduur benodigd om een product te ontwerpen totdat het op de markt verschijnt. De benodigde tijd om een product op de markt te brengen is zeer belangrijk in industrieën waar de levensduur van een product kort is. Bij een korte productlevenscyclus is het belangrijk, om winst te kunnen maken, om als eerste met het product op de markt te verschijnen.

MoSCoW-methode De MoSCoW-methode is een wijze van prioriteiten stellen in onder meer de software engineering. De eisen aan het resultaat van een project worden ermee ingedeeld. Het is een afkorting, waarvan de letters staan voor:
M - must haves: deze eisen (requirements) moeten in het eindresultaat terugkomen, zonder deze eisen is het product niet bruikbaar;
S - should haves: deze eisen zijn zeer gewenst, maar zonder is het product wel bruikbaar;
C - could haves: deze eisen zullen alleen aan bod komen als er tijd genoeg is;
W - won't haves: deze eisen zullen in dit project niet aan bod komen maar kunnen in de toekomst, bij een vervolgproject, interessant zijn.
De o's in de afkorting hebben geen betekenis

dev-stack gebruikte technologiën door een bedrijf om software te ontwikkelen. Hieronder vallen de verschillende talen, frameworks de gebruikt worden om te ontwikkelen maar ook tooling dat ondersteund bij het ontwikkelen van de software.....

DECLARATION

Put your declaration here.

Amsterdam , Februari 2022

Bas Brunink

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography *"The Elements of Typographic Style"*. `classicthesis` is available for both LaTeX and LyX:

https://bitbucket.org/amiede/classicthesis/

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

http://postcards.miede.de/