



VRIJE UNIVERSITEIT BRUSSEL

WE-DINF-6537  
PROJECT SOFTWARE ENGINEERING  
ACADEMIEJAAR 2014-2015

---

## Software Project Management Plan

---

Douglas Horemans <[dhoreman@vub.ac.be](mailto:dhoreman@vub.ac.be)>

Hannah Pinson <[hpinson@vub.ac.be](mailto:hpinson@vub.ac.be)>

Ivo Vervlimmeren <[ivervlim@vub.ac.be](mailto:ivervlim@vub.ac.be)>

Noah Van Es <[noahves@vub.ac.be](mailto:noahves@vub.ac.be)>

Pieter Steyaert <[psteyaer@vub.ac.be](mailto:psteyaer@vub.ac.be)>



2 maart 2015

---

## Versiegeschiedenis

Versie	Datum	Auteurs <sup>1</sup>	Beschrijving
0.0	05/11/2014	Hannah Pinson Ivo Vervlimmeren Noah Van Es Pieter Steyaert	Aanmaak eerste versie
0.1	18/11/2014	Hannah Pinson	Gekregen feed-back gedeeltelijk doorgevoerd planning eerste sprint aangevuld
1.0	15/12/2014	Hannah Pinson	Gekregen feed-back volledig doorgevoerd beschrijving test manager en CodeClimate planning iteratie 1 aangevuld
2.0	02/03/2015	Hannah Pinson	planning iteratie 2 aangevuld

---

<sup>1</sup>Alle versies worden nagelezen door de Document Master (Ivo Vervlimmeren)



# Inhoudsopgave

<b>Versiegeschiedenis</b>	<b>1</b>
<b>1 Introductie</b>	<b>3</b>
1.1 Inleiding en Overzicht . . . . .	3
1.2 Opleveringen . . . . .	3
1.3 Evolutie van het Software Project Management Plan . . . . .	4
1.4 Referentiemateriaal . . . . .	5
1.5 Definities en Afkortingen . . . . .	5
<b>2 Projectorganisatie</b>	<b>6</b>
2.1 Model van het softwareproces . . . . .	6
2.2 Organisatie . . . . .	7
2.3 Grenzen en raakvlakken van de organisatie . . . . .	9
2.4 Projectverantwoordelijkheden . . . . .	10
<b>3 Managementproces</b>	<b>10</b>
3.1 Beperkingen, objectieven en prioriteiten . . . . .	10
3.2 Risicomanagement . . . . .	10
3.2.1 algemene risico's . . . . .	10
3.2.2 projectgebonden risico's . . . . .	11
3.3 Opvolging, controle en communicatie . . . . .	12
<b>4 Technisch proces</b>	<b>16</b>
4.1 Technische tools en technieken . . . . .	16
<b>5 Werkpaketten en schema's</b>	<b>17</b>
5.1 Sprint 1: Inloggen en Registreren . . . . .	17
5.2 Sprint 2: Publicatie uploaden . . . . .	18
5.3 Sprint 3: Bibliotheek . . . . .	20



# 1 Introductie

## 1.1 Inleiding en Overzicht

Het doel van dit softwareproject is het ontwikkelen van een webapplicatie die het enerzijds mogelijk maakt voor onderzoekers om wetenschappelijke publicaties te beheren en die anderzijds de netwerken van deze onderzoekers analyseert en op een aantrekkelijke manier visualiseert. Deze applicatie, die de naam SKRIBL draagt, wordt binnen het kader van het opleidingsonderdeel *software engineering* gecreëerd gedurende het academiejaar 2014-2015. De rol van externe controle en klant worden hierbij vervuld door de titularis Ragnhild Van Der Straeten en de assistent Jens Nicolay van dit opleidingsonderdeel.

SKRIBL stelt gebruikers in staat om op een eenvoudige manier een lijst van eigen publicaties aan te leggen. Het is ook mogelijk publicaties van andere gebruikers en externe publicaties op te zoeken, te raadplegen en te bewaren. Bovendien kunnen alle publicaties geannoteerd worden, en kunnen ook niet-publicaties als slides en bijlagen aan de publicaties gerelateerd worden. Aan de hand van metadata analyseert SKRIBL de posities en onderlinge relaties binnen de wetenschappelijke wereld van de onderzoekers en hun publicaties. Dit laat SKRIBL enerzijds toe de meest relevante publicaties aan bepaalde gebruikers voor te stellen naargelang hun interesses en onderzoeksdomeinen aan de hand van een feedback-systeem dat hun voorkeuren leert kennen. Anderzijds zorgen deze analyses ervoor dat een gebruiker (zijn of haar eigen positie binnen) wetenschappelijke netwerken kan evalueren: aan de hand van de metadata kunnen interactieve grafen gecreëerd worden die het netwerk van de onderzoekers en de zwaartepunten binnen hun vakgebied weergeven, en gebruikers kunnen bovendien hun eigen statistieken raadplegen en visualiseren. De uiteindelijke applicatie zal zowel beschikbaar zijn via een standaard webinterface die eveneens geoptimaliseerd zal zijn voor mobiel gebruik. Daarnaast zal deze mobiele interface ook extra features ondersteunen die later in het ontwikkelingsproces zullen vastgelegd worden.

SKRIBL zal ontwikkeld worden door een team van vijf studenten in de computerwetenschappen volgens de principes van een Agile development software process. De totale werkperiode wordt opgedeeld in iteraties en sprints, waarbij het doel is een werkende partiële versie van de software te ontwikkelen en daarbij telkens een zogeheten mijlpaal te bereiken. Het uiteindelijke doel van het totale project is het opleveren van een eindproduct dat aan alle vooropgestelde eisen van de klant voldoet en dat bovendien op ieder vlak van hoge kwaliteit is. Daarnaast zullen verschillende documenten betreffende het softwareproces opgemaakt en onderhouden worden en zal een deel van het werkproces publiek worden gemaakt via een website [2]. Ten slotte zal er over de voortgang van het project ook telkens worden gecommuniceerd door middel van presentaties.

## 1.2 Opleveringen

De volgende documenten worden gedurende het project onderhouden en bij iedere iteratie opgeleverd:

- Software Project Management Plan (SPMP)
- Software Test Plan (STD)
- Software Requirements Specification (SRS)



- Software Design Document (SDD)
- Minutes van alle vergaderingen.

Aan het einde van iedere iteratie wordt ook telkens een werkende partiële versie met de bijhorende broncode, documenten en unit-tests opgeleverd zoals bepaald in [1]. Dit gebeurt via mail en valt onder de verantwoordelijkheid van de Project Manager. Daarnaast vindt er na iedere iteratie ook nog een presentatie plaats, waarvan de voorbereiding onder de verantwoordelijkheid van de Presentation Master valt. De data van opleveringen en presentaties zijn weergegeven in figuur 1.

<b>Datum</b>	<b>To Do</b>
Woensdag 05/11/2014	inleveren SPMP
Woensdag 19/11/2014	eerste versie documenten
Maandag 15/12/2014	Einde iteratie 1: opleveren code en documenten
Vrijdag 19/12/2014	presentatie
Dinsdag 03/03/2015	Einde iteratie 2: opleveren code en documenten
Woensdag 11/03/2015	presentatie
Maandag 20/04/2015	Einde iteratie 3: opleveren code en documenten
Woensdag 22/04/2015	presentatie
Vrijdag 15/05/2015	Einde iteratie 4: Finale oplevering.
Woensdag 20/05/2015	finale presentatie.

Figuur 1: Vooropgestelde data van opleveringen en presentaties.

### 1.3 Evolutie van het Software Project Management Plan

Dit Software Project Management Plan is opgesteld volgens de IEEE 1085.1-1987 standaard,[3]. In iedere iteratie zal het verder aangevuld worden, en voor iedere oplevering wordt een voorlopige officiële versie afgewerkt. In volgende versies zullen voornamelijk de onderdelen 4 (Technisch proces) en 5 (Werkpaketten en schema's) verder uitgewerkt worden. Suggesties voor toekomstige versies zijn geplaatst binnen vierkante haakjes en worden in cursief weergegeven.



## 1.4 Referentiemateriaal

- [1] R. an Der Straeten and J. Nicolay. Organisatie van het project. Technical report, Vrije Universiteit Brussel.
- [2] D. Horemans. Skribl. <http://skribl.tk>.
- [3] S. C. McConnell. Software project management plan (spmp) template. <http://users.csc.calpoly.edu/~jdalbey/205/Mgmt/SPMP.html>.
- [4] D. Wells. Agile software development: A gentle introduction. <http://www.agile-process.org>.

## 1.5 Definities en Afkortingen

- SPMP: Software Project Management Plan
- STD: Software Test Plan
- SRS: Software Requirements Specification
- SDD: Software Design Document
- RL: Risk List
- PM: Project Manager
- QA(M): Quality Assurance (Manager)
- CM: Configuration Manager



## 2 Projectorganisatie

### 2.1 Model van het softwareproces

De applicatie zal gecreëerd worden door iteratieve ontwikkeling en incrementele oplevering, gestuurd door de principes van de agile development methode [4]. Concreet wordt het volledige proces opgedeeld in vier iteraties waarvoor de uiteindelijke doelen in grote lijnen vastliggen. Deze iteraties bevatten, afhankelijk van de totale duur, één of twee sprints. De sprints bestaan uit drie fasen die mogelijk deels overlappen: een plannings- en designfase, een uitvoeringsfase, en een test- en refactorfase. Een sprint beslaat een periode van twee tot vier weken. Het doel van iedere sprint is het ontwikkelen van een werkende partiële versie en het bereiken van een zogenaamde mijlpaal:

#### **Mijlpaal 1: inloggen en registreren mogelijk**

Nieuwe gebruikers kunnen een account aanmaken en geregistreerde gebruikers kunnen inloggen na invoer van correcte gebruikersnaam en wachtwoord.

#### **Mijlpaal 2: eigen publicaties kunnen worden toegevoegd**

Gebruikers kunnen publicaties toevoegen zoals beschreven in scenario 1 [1].

#### **Mijlpaal 3: publicaties kunnen opgezocht, toegevoegd en geannoteerd worden**

Gebruikers kunnen publicaties van anderen opzoeken en bewaren zoals beschreven in scenario 3 [1]. Gebruikers kunnen publicaties annoteren en niet-publicaties (bijlagen, slides e.d.) kunnen worden gelinkt zoals beschreven in scenario 6 [1].

#### **Mijlpaal 4: metadata wordt gegenereerd en eenvoudig weergegeven**

Er worden automatisch up-to-date bibliometrische gegevens toegekend aan publicaties. De relevantie van een publicatie voor een gebruiker kan gekwantificeerd worden en een top vijf kan op deze manier worden gegenereerd (statistisch). Voor iedere gebruiker worden statistische gegevens berekend en numeriek weergegeven.

#### **Mijlpaal 5: netwerk wordt gecreëerd en gevisualiseerd**

Gebruikers en publicaties worden beschouwd als een netwerk, verbonden door standaard relaties als auteurschap, citaties, onderzoeksdomeinen ed. Aanpassingen aan dit netwerk worden consistent doorgevoerd. Het netwerk van een gebruiker kan zo dan worden weergegeven als een interactieve graaf.

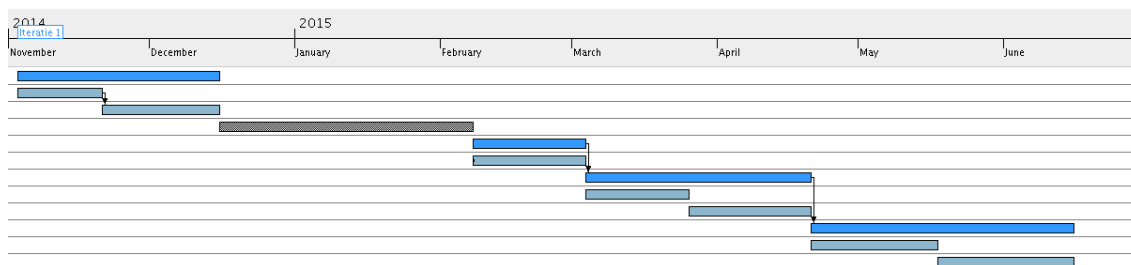
#### **Mijlpaal 6: uitgebreide interactie en visualisatie mogelijk**

De relevantie-score van een publicatie wordt aangepast aan de voorkeuren van de gebruiker. Gebruikers kunnen publicaties linken op manieren die de applicatie niet standaard voorziet. De gebruiker kan grafieken laten genereren en exporteren.



### Mijlpaal 7: mobiele interface en extra features

Naast de standaard webinterface is er ook een interface voor mobiel gebruik. Bij deze interface wordt er rekening gehouden met de specifieke mogelijkheden en beperkingen van mobiele apparaten. In een later stadium van het project zal er worden beslist welke extra functionaliteit aan deze interface zal worden toegevoegd.



Figuur 2: Gantt chart van het SKRIBL project. In blauw de geplande iteraties, in grijs de geplande sprints. Afbeelding gegeneerd via Ganttproject, <http://www.ganttproject.biz>.

Iteratie	Sprint	Begin	Einde	Mijlpaal	Oplevering
1	1	01/11/14	20/11/14	1: inloggen & registreren	nee
	2	21/11/14	15/12/14	2: eigen publicaties	ja
2	3	08/02/15	03/04/15	3: publicaties uitgebreid	ja
3	4	04/03/15	25/03/15	4: metadata	nee
	5	26/03/15	20/04/15	5: netwerk	ja
4	6	21/04/15	17/05/15	6: interactie & visualisatie	nee
	7	18/05/15	15/06/15	7: mobiel	ja

Tabel 1: Algemene planning van het ontwikkelingsproces.

De algemene planning van het project is te vinden in tabel 1 en deze planning is ook weergegeven in figuur 2. Bij deze planning worden de vooropgestelde scenario's niet chronologisch uitgewerkt. In plaats daarvan zijn scenario's of delen van scenario's gegroepeerd volgens te bereiken mijlpalen, en dit op zo'n manier dat alle scenario's aan het einde van het project geïmplementeerd zullen zijn.

## 2.2 Organisatie

De verschillende leden van de groep nemen elk de twee (of een enkele keer drie) van de hieronder vermelde rollen op zich. Dit wil zeggen dat zij de eindverantwoordelijkheid dragen voor de vermelde aspecten van het ontwikkelingsproces. Daarnaast fungeert ieder lid ook voor twee of drie rollen als back-up persoon om opvolging en continuïteit binnen het project te verzekeren, en om de hoofdverantwoordelijken bij te staan. Ten slotte is ieder lid ook developer en schrijft hij/zij dus ook de unit testen voor de onderdelen die hij/zij zelf ontwikkelt.





### **Project Manager**

- algemene coördinatie en planning
- taakverdelingen
- vastleggen, voorbereiden en leiden van vergaderingen
- onderhouden van het SPMP
- communicatie met end-user

### **Requirements Manager**

- analyseren van requirements: bijvoorbeeld informatie over journals en publicaties, documentformaten, enz.
- Het nauwkeurig oplijsten van requirements, op zo'n manier dat design en testing mogelijk is.
- Het onderhouden van het SRS.

### **Configuration Manager**

- bepalen afspraken en conventies metasoftware en tools (e.g., github structuur, conventies voor code commenting)
- waken over consistentie en integratie van verschillende software onderdelen (in samenspraak met de software architect)

### **Software Architect**

- algemene architectuur van de software bepalen
- ontwerpen API's van de modules
- onderhouden van het SDD

### **Quality Assurance Manager**

- algemene toezicht op kwaliteit van de code en de ontwikkelde software

### **Test Manager**

- verzamelen en inspecteren van de door de leden geschreven tests
- onderhouden van STD

### **Database Manager**

- design en onderhoud database



**Webmaster**

- (technisch) design en onderhoud van websites

**Design en front-end**

- (creatief) design en ontwikkeling van GUI's en datavisualisaties

**Document Master**

- overzicht en specificatie van documenten die opgeleverd moeten worden, in samenspraak met specifieke verantwoordelijke
- consistentie van de lay-out bewaken
- minutes van vergaderingen opstellen

**Presentation Master**

- voorbereiding van structuren, slides en eventueel demo's voor de presentaties

## 2.3 Grenzen en raakvlakken van de organisatie

De titularis en assistent van het opleidingsonderdeel software engineering, Ragnhild Van Der Straeten en Jens Nicolay respectievelijk, vervullen samen enerzijds de rol van externe controle en anderzijds de rol van klant.

Om hen inzicht in het werkproces te geven onderhoudt de groep een website met een requirements dashboard en links naar de github repository en opgeleverde documenten. Deze website valt onder de verantwoordelijkheid van de webmaster.

Aan het eind van elke iteratie worden de gevraagde bestanden en documenten opgeleverd en kort na iedere iteratie stelt de groep de voortgang en bereikte resultaten voor tijdens een presentatie. De communicatie en de oplevering valt onder de verantwoordelijkheid van de Project Manager. Het voorbereiden en in goede banen leiden van de presentaties is de taak van de Presentation Master.



## 2.4 Projectverantwoordelijkheden

Rol	Lead	Back-up
Project Manager	Hannah	Pieter
Configuration Manager	Pieter	Douglas
Requirements Manager	Hannah	Noah
Software Architect	Noah	Ivo
Quality Assurance Manager	Noah	Pieter
Test Manager	Douglas	Ivo
Database Manager	Ivo	Hannah
Webmaster	Douglas	Hannah
Design en front-end	Pieter	Douglas
Document master	Ivo	Noah
Presentation master	Douglas	Noah

Tabel 2: Verantwoordelijkheden binnen het project: lead en back-up rollen.

## 3 Managementproces

### 3.1 Beperkingen, objectieven en prioriteiten

De baseline van het management proces is als volgt samen te vatten: eerst en vooral staat het verwezenlijken van prioritaire doelen en bijhorende hoge kwaliteit voorop; het streven naar resultaten met een lage prioriteit gebeurt na samenspraak met de groep en enkel als ieder lid de taak voor zichzelf haalbaar vindt. De project manager verbindt zich ertoe dit voor ieder lid afzonderlijk op te volgen. De Quality Assurance Manager waakt verder over de kwaliteit van de behaalde doelen.

### 3.2 Risicomanagement

Risico's zullen op volgende manieren gedetecteerd en opgelijst worden:

- door het nagaan van de risico's voor andere, gelijkaardige softwareprojecten
- door iedere openingsvergadering enkele minuten aan het onderwerp te besteden
- door het aandragen van verwachte risico's door leden aan de PM

De risico's zullen opgevolgd worden door de PM.

#### 3.2.1 algemene risico's

Een te grote werklust

- waarschijnlijkheid: groot



- melding: Groepsleden geven aan wanneer de werklast voor hen te groot dreigt te worden.
- oplossing: Aan het begin van iedere sprint worden de opdrachten geprioriteerd. Indien de werkdruk te hoog zou worden kunnen minder belangrijke opdrachten weggelaten of opgeschoven worden. De laatste sprint laat ruimte voor het uitvoeren van eventueel onafgewerkte features. Tijdens blok- en examenperiodes en tijdens vakanties wordt er niet verwacht van de groepsleden aan het project te werken.

#### **Onvoldoende ervaring of kennis**

- waarschijnlijkheid: groot
- melding: Groepsleden geven duidelijk aan wanneer zij het gevoel hebben dat het hen aan de nodige kennis of ervaring ontbreekt. De Project Manager maakt ruimte in de planning voor het wegwerken van de achterstand (indien mogelijk) en geeft dit ook expliciet als taak op.
- oplossing: Het herhalen van cursussen en/of het volgen van online tutorials.

#### **Afwezigheid of ziekte**

- waarschijnlijkheid: groot
- melding: Groepsleden laten zo vroeg mogelijk weten wanneer en hoelang zij afwezig zullen zijn.
- oplossing: Voor iedere rol is er een persoon aangeduid als back-up. Deze volgt de activiteiten van de hoofdverantwoordelijke op en kan deze rol overnemen indien nodig.

#### **Eindresultaat is niet gebruiksvriendelijk**

- waarschijnlijkheid: gemiddeld
- melding: De beta versies zullen worden voorgelegd aan enkele externe testers, zoals bereidwillige assistenten, maar ook personen die niet geaffilieerd zijn aan de wetenschappelijk wereld.
- oplossing: De feedback van deze testers wordt gebruikt om aanpassingen aan het product door te voeren

### **3.2.2 projectgebonden risico's**

#### **vorm (meta)data niet voldoende consistent**

- waarschijnlijkheid: groot
- beschrijving: de door gebruikers ingevoerde data is niet van een consistente vorm, e.g. de ene gebruiker geeft als universiteit 'VUB', de andere 'Vrije Universiteit Brussel'; de ene geeft auteurs met initialen, de andere met voornamen voluit etc. Dit bemoeilijkt het vinden van relaties om het netwerk van een gebruiker op te bouwen.
- oplossing: Waar mogelijk het verwachte formaat specificeren of de keuzemogelijkheden beperken; eventueel reeds in het systeem aanwezige data als optie voorstellen.



### 3.3 Opvolging, controle en communicatie

Per sprint worden standaard twee vergadermomenten voorzien waarbij het hele team aanwezig is: een openingsvergadering en een follow-up vergadering. Dit komt neer op drie tot vier sprintbijeenkomsten per maand. De openingsvergadering vindt plaats aan het begin van de sprintperiode. De follow-up vergadering vindt plaats tussen midden en eind van de sprintperiode, d.w.z. aan het eind van de uitvoeringsfase en in het begin van de refactor- en testfase.

De openingsvergadering bestaat uit de volgende componenten:

- deel 1 (ong. 45 min.):

- opening en overlopen agendapunten
- micropresentaties (belichten de aspecten van het ontwikkelingsproces waarvoor het betreffende groepslid verantwoordelijk is)
- variabele agendapunten en opmerkingen, zoals op te leveren documenten, afspraken e.d.

-deel 2 (ong. 45 min):

- design en planning van de sprint
- taakverdelingen
- overlopen van risico's

De follow-up vergadering bestaat uit volgende componenten:

- (ong. 30 min.):

- opening en overlopen agendapunten
- micropresentaties en reeds uitgevoerde taken
- aanpak van quality assurance en testing
- overlopen van plan voor configuratie, integratie en eventueel oplevering
- verwachte problemen
- eventuele (her)verdeling van taken en nieuwe acties

Naast de sprintbijeenkomsten kunnen er bijeenkomsten plaatsvinden tussen de groepsleden onderling, om deelaspecten van het project te bespreken of om bijvoorbeeld presentaties voor te bereiden. Daarnaast kan de QA Manager ook een korte reviewsessie organiseren met de eindverantwoordelijke van een bepaald onderdeel mocht de kwaliteit ervan betwist worden.

Doorlopende en overzichtelijke communicatie wordt verzekerd door het gebruik van enkele hiervoor ontwikkelde tools en communicatieplatformen. De (huidige) selectie van communicatiemiddelen werd na overleg binnen het volledige team gekozen uit een lijst met voorstellen samengesteld door de configuration manager. De keuzes werden geïnspireerd door de eerdere ervaring van teamleden binnen kleinere softwareprojecten en startups.



**Algemene communicatie & persoonlijke communicatie : Slack**

Slack is een zeer eigentijds communicatieplatform, veelal gebruikt door nieuwe startups. Mededelingen over verschillende onderwerpen worden gegroepeerd in zogenoemde kanalen. Bovendien biedt Slack een waaier van integraties aan voor andere development- en communicatiediensten. Skribl gebruikt op dit moment 7 kanalen met individuele integraties naar Blossom, Github, Trello, Google Docs en Google Hangouts.

Als alternatieven werden mail, Facebook, Google Groups, IRC, Skype en BaseCamp onderzocht. Geen van deze platformen biedt een degelijke uitgebreide integratie met third-parties aan, hetgeen de doorslag gaf voor Slack.



### Algemeen takenbeheer : Trello



Deze applicatie stelt een gebruiker en/of team in staat om lijstjes aan te maken met entries die gemakkelijk van lijstje kunnen verplaatst worden. Elke entry kan gekoppeld worden aan een of meerdere teamgenoten, kan een label krijgen, een deadline, etc. Deze tool wordt gebruikt voor alle task management die niet rechtstreeks met de development van de applicatie te maken heeft. Als alternatief werd Kanbanize besproken, maar vanwege de sterke integratie met Slack werd toch voor Trello geopteerd.

### Opvolging agile development: Blossom



Blossom is een project management tool, specifiek ontwikkeld voor agile software processen. Met deze applicatie is het mogelijk lijsten met te ontwikkelen features op te stellen, aan deze features ontwikkelaars en deadlines toe te wijzen, en vervolgens de status van het ontwikkelingsproces aan te passen en op te volgen. Bovendien geeft Blossom ook per feature performantiestatistieken (cycle time performance analytics) weer.

Blossom is een betalende applicatie. Na overleg met een van de founders van Blossom werd echter aan de leden van SKRIBL voor academische doeleinden het gratis gebruik ervan toegestaan tot het einde van het academiejaar.

Het team vond het zinvol om buiten Trello een gerichte tool te gebruiken voor het agile development proces. Voor Blossom is er een integratie aanwezig met GitHub en Slack, waardoor deze management tool de voorkeur van het team wegdroeg.



**Beheren van documenten: Google Drive & ShareL<sup>A</sup>T<sub>E</sub>X**

Door gemeenschappelijk in de cloud te werken met alle services van Google Drive (Docs, Spreadsheets of Presentaties) kunnen documenten collectief en in real-time aangepast en beheerd worden. Op deze manier wordt duplicatie vermeden en heeft iedereen steevast de laatste versie van een werkdokument. Documenten waarvan de inhoud afgewerkt is, worden door de document master(s) voor oplevering van een consistente en academische lay-out voorzien door gebruik van ShareL<sup>A</sup>T<sub>E</sub>X. Als alternatief werd Evernote beschouwd, maar deze biedt niet dezelfde collectieve en real-time features als Google Drive.

**Automatische code review: CodeClimate**

CodeClimate is een tool die, eens gekoppeld aan github, automatisch javascript code inspecteert en vervolgens rapporten genereert over de stijl en kwaliteit van deze code. CodeClimate biedt op die manier een objectieve analyse van de geschreven broncode. De teamleden kunnen de feedback van CodeClimate gebruiken om de kwaliteit van hun code te verbeteren, nog voor de Quality Assurance Manager deze hoeft te inspecteren.





## 4 Technisch proces

### 4.1 Technische tools en technieken

Voor de ontwikkeling van SKRIBL wordt enkel gebruik gemaakt van JavaScript, HTML5 en CSS, en de daarbij horende open-sources frameworks en bibliotheken die nodig worden geacht. GitHub, geïntegreerd met Slack en Blossom, fungeert als repository en gedeeltelijk als collaboration tool. Onder leiding van de configuration manager wordt het ingezet als platform voor version-control, issue tracking en reviewing. Merging-issues en bugs worden volgens de standaard methoden van het platform behandeld.

De conventies voor documentatie en code commenting worden opgesteld door de configuration manager (CM) en de quality assurance manager (QAM). Hierbij wordt gebruikt gemaakt van JS-Docs.

De test manager (TM) verzamelt de unit testen geschreven door de afzonderlijke teamleden, en houdt het overzicht over het gehele testproces. Hierbij wordt gebruik gemaakt van het jasmine testing framework. Meer informatie hieromtrent is te vinden in het STD.

De quality assurance manager zal verder defecten en inconsistenties opsporen om de kwaliteit van de ontwikkelde onderdelen te garanderen. Dit zal gebeuren door:

- de kwaliteit van de geschreven op te volgen aan de hand van CodeClimate
- na te gaan of alle modules correct worden geïmplementeerd zoals omschreven in het SDD
- te controleren dat aan alle vooropgestelde requirements voldaan is (zoals opgelijst in het SRS)

Automatische code-inspectie wordt uitgevoerd door de CodeClimate tool. Een door CodeClimate gerapporteerde A of B wordt door het team als voldoende beschouwd. Indien de code van lagere kwaliteit is (of wanneer de QAM andere significante problemen vaststelt) wordt de code bijgeschaafd door het desbetreffende teamlid, eventueel volgens de richtlijnen van de QAM.

Meer informatie over de syseemomgeving, de gebruikte database e.d. is te vinden in het SDD.



## 5 Werkpaketten en schema's

*[Dit onderdeel zal na de opening van iedere sprint verder aangevuld worden.]*

De werkpaketten en planningen voor iedere sprint zullen weergegeven worden in eenvoudige Gantt charts. De lengte van de balkjes op het diagram stemt hierbij overeen met de verwachte werklust voor elke opdracht. 1 eenheid op het diagram komt overeen met 1 dag, waarbij 1 dag beschouwd als de gemiddelde werkdag van een student. Aangenomen wordt dat er op deze dag ruwweg twee uur aan de opdracht kan gespendeerd worden. De verwachte werklust wordt bepaald op de openingsvergadering van de sprint door een eenvoudig systeem van stemming en gemiddeldes.

### 5.1 Sprint 1: Inloggen en Registreren

Figuur 3 geeft de planning voor de eerste sprint weer. Deze sprint loopt van 1 november 2014 (notatie 11/01/14) tot 20 november 2014 (notatie 11/20/14). Deze sprint heeft als mijlpaal 'mijlpaal 1: inloggen en registreren' en implementeert volgende features:

- registratie
- login/logout
- accountbeheer
- taal

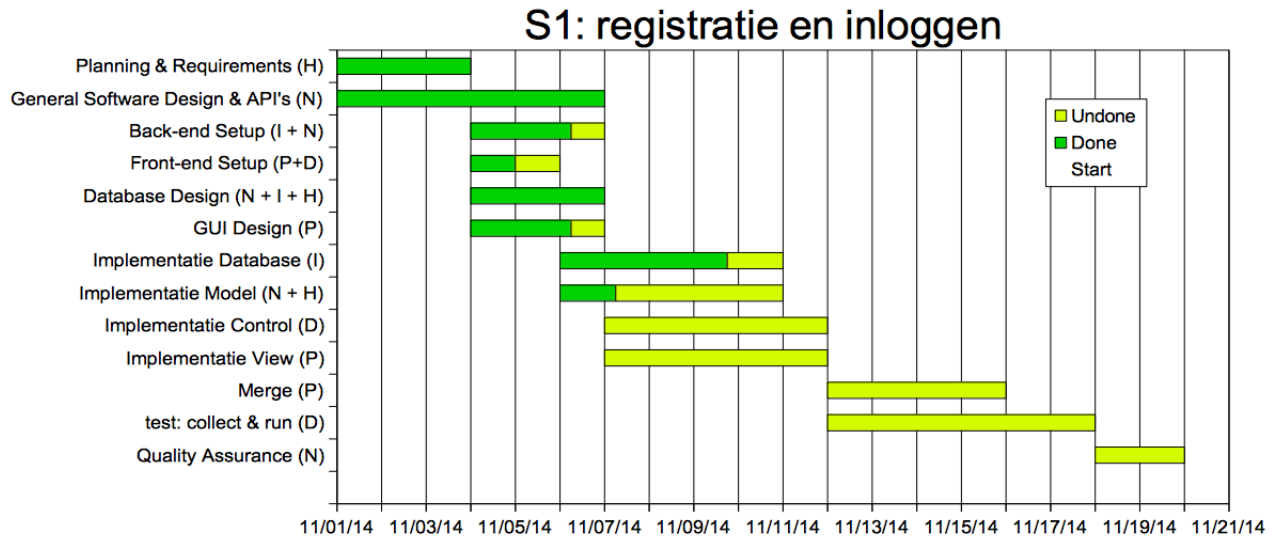
De bijhorende requirements zijn te vinden in het Software Requirements Specification (SRS) document.

Zoals op figuur 3 is af te lezen, is er op datum 11/18/14 en grote achterstand in het ontwikkelingsproces. Deze achterstand heeft vier oorzaken:

- het uitzoeken en installeren van de benodigde bibliotheken en software duurde langer dan verwacht
- een van de teamleden was enkele dagen ziek
- het ontbrak enkele teamleden aan de juiste kennis, onder meer in het gebruik van JavaScript, waardoor er extra tijd nodig was om deze zaken bij te spijkeren
- taken die niet in deze planning zijn opgenomen, onder meer het schrijven van de op te leveren documenten, bleken zeer tijdrovend te zijn

De PM zal deze situatie opvolgen en trachten te remediëren. In samenspraak met het team zullen enkele minder prioritaire requirements uitgesteld worden.





Figuur 3: Gantt chart voor sprint 1, op datum 11/18/14 (18 november 2014), einde van de sprint. Links van het diagram een korte beschrijving van de taak en de eerste letter van de voornaam van het corresponderende teamlid. Figuur gecreëerd met OpenOffice Calc.

## 5.2 Sprint 2: Publicatie uploaden

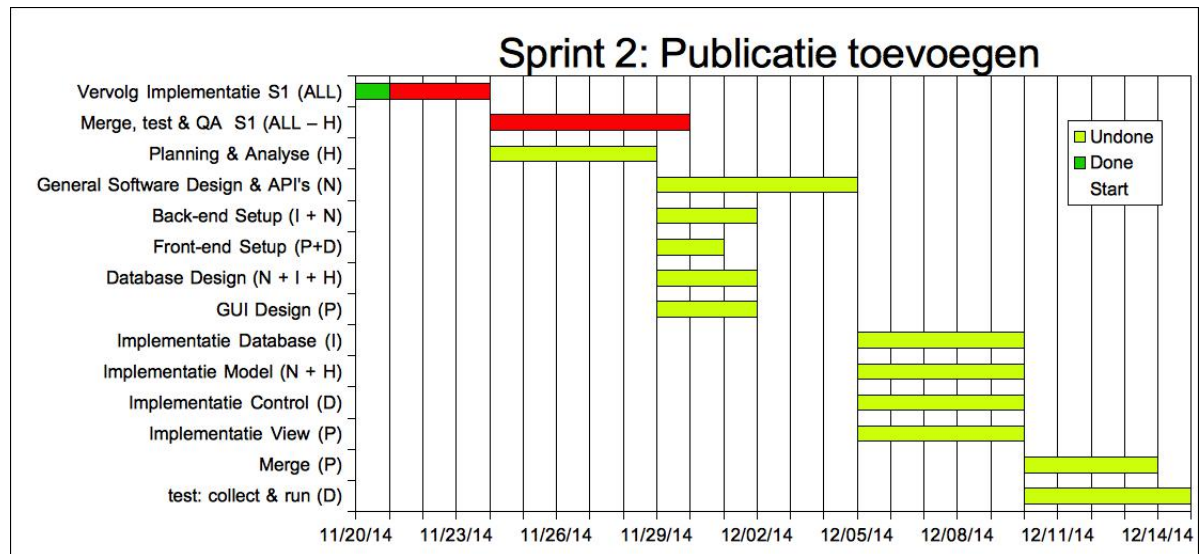
Figuur 4 geeft de planning voor de tweede sprint weer, inclusief remediëring voor de achterstand opgelopen tijdens sprint 1. Sprint 2 loopt van 20 november 2014 (notatie 11/20/14) tot 15 december 2014 (notatie 12/15/14). Deze sprint heeft als mijlpaal 'mijlpaal 2: publicatie toevoegen' en implementeert volgende features:

- toevoegen van publicatie in PDF of Bibtex formaat + metadata extractie en manuele aanvulling

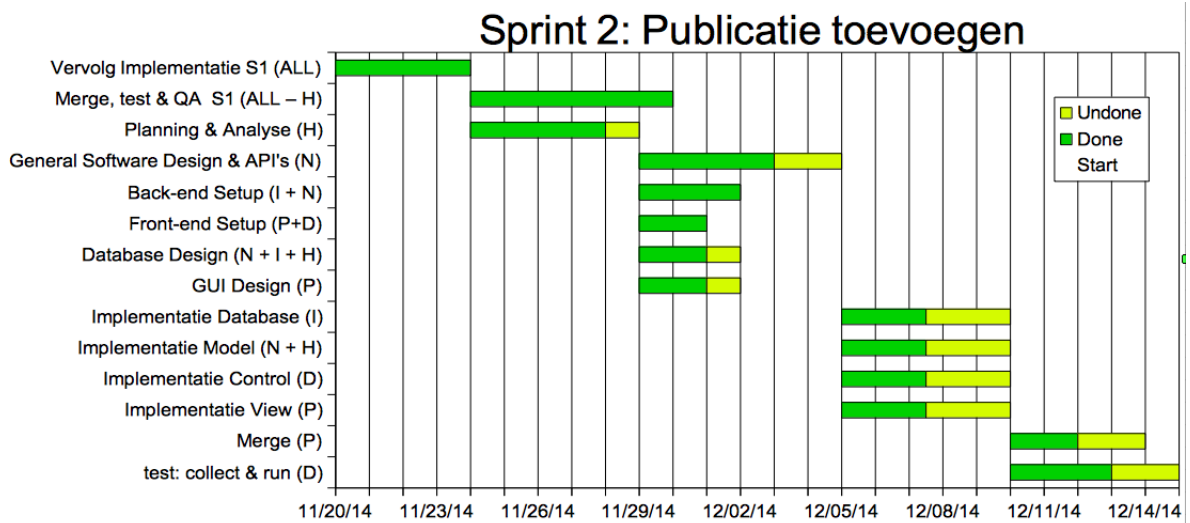
De bijhorende requirements zijn te vinden in het Software Requirements Specification (SRS) document.

Aan het einde van de sprint 2, datum 20 november, is niet aan alle hoog en gemiddeld geprioriteerde requirements voor sprint 2 voldaan. Zoals te zien in figuur 5 is sprint 1 echter volledige afgerond en is de daarbij opgelopen achterstand grotendeels weggewerkt.





Figuur 4: Gantt chart voor sprint 2, op datum 20 november 2014, begin van de sprint. De rode balkjes corresponderen met de weg te werken achterstand in sprint 1. Figuur gecreëerd met OpenOffice Calc.

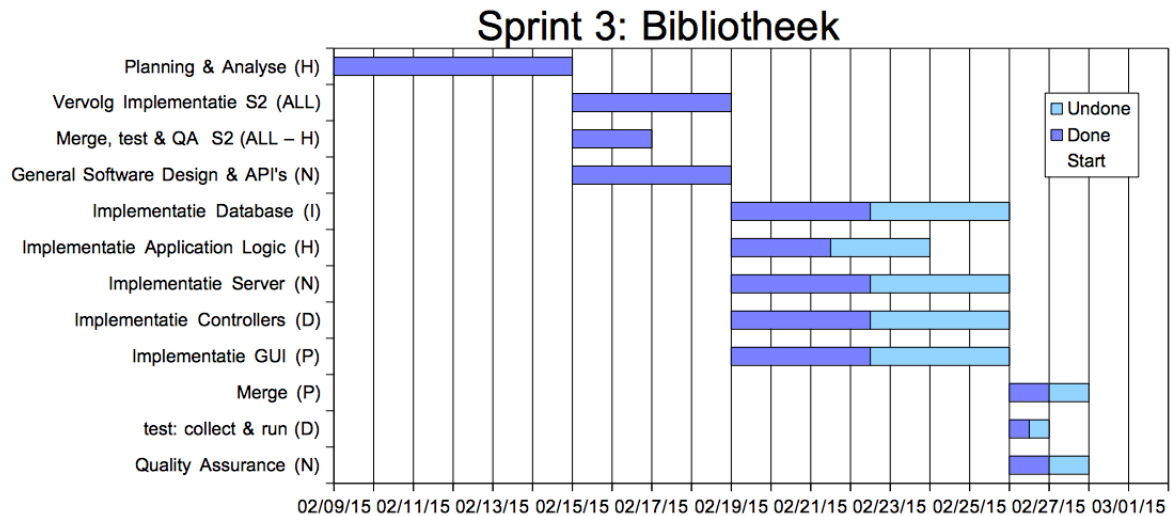


Figuur 5: Gantt chart voor sprint 2, op datum 15 december 2014, einde van de sprint. Figuur gecreëerd met OpenOffice Calc.



### 5.3 Sprint 3: Bibliotheek

Figuur 6 geeft de planning voor de implementatie van de derde sprint weer, inclusief remediëring voor de achterstand opgelopen tijdens sprint 2. Sprint 3 loopt van 9 februari 2015 tot 2 maart 2015, en heeft als mijlpaal 'mijlpaal 3: bibliotheek' (zie SRS). De werkpaketten, deels gebaseerd op deze Gantt chart, zijn weergegeven in 3. Figuur 7 ten slotte geeft een samenvatting van de reële werkuren voor sprint 3. De geplande werklast was (met opzet) te hoog; dit blijkt ook uit het feit dat de doelstellingen voor deze sprint slechts gedeeltelijk behaald zijn. Volgende planningen zullen opgesteld worden met als richtgetal 30u per persoon per sprint, of ongeveer 10u per persoon per week.

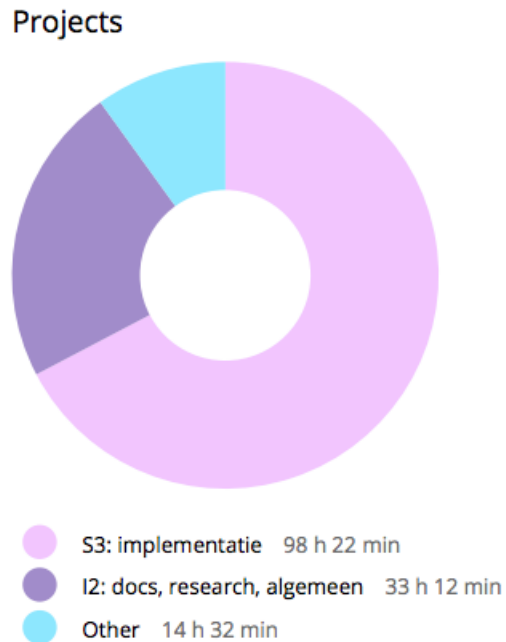


Figuur 6: Gantt chart voor sprint 3, op datum 02 maart 2015, einde van de sprint. Figuur gecreëerd met OpenOffice Calc.



Type	Berekening duur	Geplande duur	Reële duur aan einde sprint
Taken algemeen: planning, documenten, etc.	planning en analyse (10u) + 4u per doc + 1u pp overig	10+16+5 = 31u	±33u
Bijscholing & tutorials	2u pp	10u	±5 u
Vergadering	2 maal 1u pp	10u	±8 u
Implementatie, zie fig. 6	2u per eenheid * aantal toegewezen teamleden	133u	±100u (implementatie onvolledig)
Totaal		±185u, 37u pp	±150u, 30u pp (via Toggl)

Tabel 3: Geplande en reële werkpaketten sprint 3.



Figuur 7: Pie chart met reële werkpaketten voor sprint 3, op datum 02 maart 2015, einde van de sprint. Figuur gecreëerd via Toggl.

