



## Professionele Bachelor Toegepaste Informatica



**FIREWOLF**  
STUDIOS

## Softwaremanagement in een gamedevelopmentomgeving

Lien Peeters

Promotoren:

Frank Van Geirt  
Lowie Vangaal

Firewolf Studios  
Hogeschool PXL Hasselt



---

**Bachelorpaper Academiejaar 2017-2018**





## Professionele Bachelor Toegepaste Informatica



**FIREWOLF**  
STUDIOS

## Softwaremanagement in een gamedevelopmentomgeving

Lien Peeters

Promotoren:

Frank Van Geirt  
Lowie Vangaal

Firewolf Studios  
Hogeschool PXL Hasselt



---

**Bachelorpaper Academiejaar 2017-2018**

## Dankwoord

Allereerst wil ik Firewolf Studios en in het bijzonder Frank Van Geirt bedanken om mij de kans te geven mijn stage aan te vatten bij hun als student softwaremanagement. Ik kon bij hem altijd terecht met vragen of bedenkingen.

Daarnaast wil ik ook graag mijn collega's op mijn stageplaats bedanken. Wouter Spaas wil ik bedanken om altijd klaar te staan om feedback te geven op mijn werk of om te luisteren naar mijn vragen en opmerkingen. En Stijn Mommen voor zijn uitleg en medewerking in verband met de projecten in de studio.

Ik wil ook graag mijn hogeschoolpromotor Lowie Vangaal bedanken voor de grondige opvolging van mijn stage. Hij was altijd snel beschikbaar bij vragen en ik heb veel gehad aan zijn grondige evaluaties van mijn werk.

Graag bedank ik ook mijn oud-collegastudenten Boyen Vaesen, Brecht Morrhey, Stijn Stas en Marit van de Moosdijk om altijd klaar te staan met tips en feedback omtrent mijn stageperiode en mijn eindwerk.

Lien Peeters

## Abstract

Firewolf Studios is een game- en animatiestudio. Ze focussen zich op het ontwikkelen van ‘serious games’. Deze serious games richten zich op het trainen van werknemers, informeren van klanten of ze bieden educatieve informatie. Ze zijn op die manier dus uniek in het ontwikkelen van serieuze games in de plaats van de vaker voorkomende games gericht op entertainment. Ze zorgen er voor dat hun games informatief zijn, maar toch fijn zijn om te spelen.

Dit bedrijf is momenteel nog in een start-upfase, waardoor projectmanagement vooral mondeling verloopt en waardoor de testprocedure nog maar minimaal ontwikkeld is. Uit deze probleemstelling is de stageopdracht ontstaan en is dit document tot stand gekomen. Binnen de stageopdracht is er getracht een oplossing te bieden voor deze twee verbeterpunten.

Het eerste verbeterpunt is het overzicht op huidige projecten. Er is geen duidelijk zichtbaar overzicht van de huidige vooruitgang binnen de projecten of van de nog af te werken taken voor de lopende projecten. De teamleden weten allemaal hoe ver ieder project ervoor staat, maar buiten papieren checklists wordt deze informatie nergens bijgehouden. Er is geen duidelijk visuele voorstelling van de voortgang van een project en het is moeilijk om in te schatten hoe veel tijd nieuwe projecten of taken in beslag zullen nemen of hoeveel tijd er nog effectief nodig is voor een project afgerond kan worden. Als oplossing hiervoor, zijn er agile methodologieën toegepast binnen het bedrijf. Ook is hieruit een onderzoeksvraag ontstaan. Er is onderzoek gebeurd naar de toepassing van issuetracking- en projectmanagement tools binnen de studio.

Om de testprocedure te verbeteren wordt er gezocht naar testmethodes binnen gamedevelopment. Ook wordt er geanalyseerd welke testmethodes het bedrijf kan toepassen om hun testprocedure te verbeteren op een manier die bij hun werkwijze past. Er is onderzocht op welke manieren men testen kan uitvoeren binnen de ontwikkelingsomgeving naar keuze van het bedrijf: Unity. Er zijn testen uitgevoerd met Unium en Jasmine, Python en PyAutoGUI en de mogelijkheid van performancetesten is bekeken. Ook is er bekeken hoe het bedrijf de informatie uit deze testen kan documenteren.

# Inhoudsopgave

Dankwoord .....	ii
Abstract .....	iii
Lijst van gebruikte figuren .....	vii
Lijst van gebruikte tabellen .....	viii
Lijst van gebruikte begrippen .....	ix
Lijst van gebruikte afkortingen.....	xi
Inleiding .....	1
I. Stageverslag.....	2
1. Bedrijfsvoorstelling.....	2
1.1 Algemene bedrijfsgegevens .....	2
1.2 Bedrijfsstructuur.....	2
1.3 Klanten en projecten .....	3
2. Voorstelling stageopdracht .....	4
2.1 Probleemstelling.....	4
2.2 Doelstelling.....	4
3. Uitwerking stageopdracht.....	5
3.1 Beschrijving aanpak.....	5
3.2 Softwaretesten binnen gamedevelopment .....	5
3.2.1 Vergelijking gametesten en traditionele softwaretesten .....	5
3.2.2 Gekende testmethodes binnen gamedevelopment .....	6
3.2.3 Gamedevelopment en testen binnen Firewolf Studios.....	7
3.2.4 Nieuwe vormen van testen binnen Firewolf Studios .....	9
3.2.5 Niet-geïmplementeerde testmethodes.....	15
3.2.6 Resultaten.....	15
3.3 Projectmanagement binnen gamedevelopment .....	16
3.3.1 Huidige situatie binnen Firewolf Studios.....	16
3.3.2 Agile binnen gamedevelopment .....	17
3.3.3 Implementaties .....	19
3.3.4 Resultaten.....	21
3.4 Bijkomende taken.....	21
3.4.1 Meewerken aan een nieuw project: Firewolf Game Builder .....	21
3.4.2 Overige brainstormsessies .....	23
3.5 Conclusie stageopdracht .....	23
4. Reflectie stageopdracht .....	24

II. Onderzoekstopic.....	25
1. Onderzoeksvraag.....	25
1.1 Omschrijving.....	25
1.2 Link met stageopdracht.....	25
2. Onderzoeksmethode.....	25
2.1 Aanpak.....	25
2.2 Analyse .....	26
3. Literatuurstudie.....	26
3.1 Aanleiding tot literatuurstudie.....	26
3.2 Paper 1: <i>Game development software engineering process lifecycle: a systematic review</i> door S. Aleem, L. F. Capretz en F. Ahmed .....	26
3.2.1 Overzicht.....	26
3.2.2 Bespreking .....	27
3.2.3 Besluit.....	28
3.3 Paper 2: <i>How are agile methods and practices deployed in video game development? A survey into Finnish game studios</i> door J. Koutonen en M. Leppänen.....	28
3.3.1 Overzicht.....	28
3.3.2 Bespreking .....	28
3.3.3 Besluit.....	29
3.4 Paper 3: <i>Is agile out there? Agile practices in gamedevelopment</i> door F. Petrillo en M. Pimenta .....	30
3.4.1 Overzicht.....	30
3.4.2 Bespreking .....	30
3.4.3 Besluit.....	30
3.5 Vergelijking en conclusie.....	30
4. Resultaten.....	31
4.1 Selectie van tools.....	31
4.1.1 Requirements .....	31
4.1.2 Onderzoek naar mogelijke tools.....	33
4.1.3 Evaluatieformulier .....	33
4.1.4 Keuze tools .....	34
4.2 Resultaten implementatie.....	35
4.2.1 YouTrack.....	35
4.2.2 JIRA .....	42
4.2.3 SpiraTeam.....	48
4.3 Vergelijking.....	52

5. Evaluatie .....	53
5.1 Meerwaarde onderzoek .....	53
5.2 Bespreking resultaten.....	53
5.2.1 Vergelijking met andere bronnen .....	53
5.2.2 Relevantie met stageopdracht .....	53
5.2.3 Aanbevelingen .....	54
5.2.4 Volgende onderzoeksvraag .....	54
5.3 Persoonlijke reflectie.....	54
Conclusie .....	55
6. Bibliografie.....	56
Bijlagen .....	57



## Lijst van gebruikte figuren

Figuur 1: Organogram van de bedrijfsstructuur van Firewolf Studios

Figuur 2: Voorbeeldquery in Unium en resulterende JSON data

Figuur 3: Voorbeeldscript gebruikmakend van Unium en Jasmine

Figuur 4: Voorbeeld van de uitvoering van een test in Jasmine

Figuur 5: PyAutoGUI tool voor Voltaire

Figuur 6: Voorbeeld van een test voor Voltaire met PyAutoGUI

Figuur 7: Unity Profiler

Figuur 8: Meetwaarden binnen GameBench

Figuur 9: Het waterfall model toegepast op Firewolf Studios

Figuur 10: Kanban toegepast op Firewolf Studios in eerste fase

Figuur 11: Kanban voorstelling post-it toegepast binnen Firewolf Studio

Figuur 12: MoSCoW diagram van de vereisten

Figuur 13: Een eerste project in YouTrack

Figuur 14: Vergelijking eigenschappen van issues in een scrum, kanban en default project in YouTrack

Figuur 15: Voorbeeld van een issue binnen YouTrack

Figuur 16: Fragment uit de lijst van issues binnen YouTrack

Figuur 17: Zoeken naar issues binnen YouTrack

Figuur 18: Vragenlijst voor een soort project in Jira en antwoord hierop

Figuur 19: Mogelijke soorten projecten binnen Jira

Figuur 20: Een issue in Jira

Figuur 21: Incidenten binnen SpiraTeam

## Lijst van gebruikte tabellen

Tabel 1: Gekende testmethodes binnen gamedevelopment

Tabel 2: Soorten testen binnen Firewolf Studios

Tabel 3: Bespreking toepassing van gekende testmethodes binnen gamedevelopment

Tabel 4: Voor- en nadelen van Unium

Tabel 5: Voor- en nadelen van PyAutoGUI

Tabel 6: Uitleg waterfall model toegepast op Firewolf Studios

Tabel 7: De vereisten van het bedrijf

Tabel 8: Planning implementatie tools

Tabel 9: Uitvoering implementatie tools

Tabel 10: Aanwezigheid requirements YouTrack

Tabel 11: Feedback eindgebruikers YouTrack

Tabel 12: Scores per categorie YouTrack

Tabel 13: Toelichting scores per categorie YouTrack

Tabel 14: Aanwezigheid requirements Jira

Tabel 15: Feedback eindgebruikers Jira

Tabel 16: Scores per categorie Jira

Tabel 17: Toelichting scores per categorie Jira

Tabel 18: Aanwezigheid requirements SpiraTeam

Tabel 19: Feedback eindgebruikers SpiraTeam

Tabel 20: Scores per categorie SpiraTeam

Tabel 21: Toelichting scores per categorie SpiraTeam

Tabel 22: Vergelijking aanwezigheid requirements voor alle tools

Tabel 23: Vergelijking scores van eindgebruikers voor alle tools

Tabel 24: Vergelijking scores op basis van categorie

Tabel 25: Vergelijking van de tools, aantal requirements versus gebruikersscore

Tabel 26: Prijsvergelijking tools

## Lijst van gebruikte begrippen

Begrip	Verklaring
<b>Continuous integration</b>	Bij <i>continuous integration</i> wordt ervoor gezorgd dat code tijdens een project continue wordt getest.
<b>Unittesten</b>	Bij <i>unittesten</i> worden stukken code afzonderlijk van elkaar getest. Voor een bepaald stuk code (een unit) worden één of meer testen geschreven. Ideaal is wanneer alle testcases onafhankelijk zijn van andere tests.
<b>Software library</b>	Een <i>software library</i> of softwarebibliotheek is een verzameling van hulpbronnen bestaande uit andere software, die een programmeur kan gebruiken bij het ontwikkelen van zijn eigen programma's.
<b>Interface</b>	Een <i>interface</i> is een koppeling tussen computersystemen. Via een interface kan er gecommuniceerd worden tussen twee systemen.
<b>Spec</b>	Een <i>spec</i> is de naam die Jasmine gebruikt om een test te benoemen.
<b>Failures</b>	Een <i>failure</i> of iets dat faalt, is een term die binnen Jasmine gebruikt wordt om aan test die faalt te benoemen.
<b>Waterfall</b>	<i>Waterfall</i> is een softwareontwikkelingsmethode. Het beschrijft hoe software ontwikkeld kan worden volgens een aantal lineaire fases die slechts in één richting doorlopen kunnen worden.
<b>Cross-functioneel</b>	Een <i>cross-functioneel</i> team is een ander woord voor een multidisciplinair team. Dit is een team bestaande uit leden die allemaal een eigen expertise of rol hebben.
<b>Milestones</b>	Een <i>milestone</i> is een term die binnen projectmanagement wordt gebruikt om specifieke punten of mijlpalen binnen een project aan te duiden.
<b>Best practices</b>	<i>Best practices</i> beschrijven werkmethodes of technieken waarvan bewezen is dat ze effectiever zijn dan andere methoden of technieken.
<b>Feature</b>	Een <i>feature</i> is een ander woord om een eigenschap van ontwikkelde software of software in ontwikkeling te benoemen.
<b>Bug</b>	Een <i>bug</i> is een fout in een programma dat er voor zorgt dat het programma niet functioneert zoals het hoort.
<b>Botten</b>	<i>Botten</i> of <i>botting</i> is een term die gebruikt wordt binnen de game-industrie om aan te duiden dat het spelen van een game geautomatiseerd wordt door een eindgebruiker.
<b>Use case</b>	Een <i>use case</i> is een beschrijving van het gedrag van een systeem of software. Het beschrijft welke gebruiker (wie), wat kan doen binnen een systeem.
<b>Mock-up</b>	Een <i>mock-up</i> is een ontwerp gemaakt om de te ontwikkelen schermen binnen een softwareproduct, voor te kunnen stellen.
<b>Auto-completion</b>	<i>Auto-completion</i> is een eigenschap die bepaalt dat een applicatie kan raden wat de rest van een woord is dat de gebruiker aan het typen is.

<b>Workflow</b>	Een <i>workflow</i> of werkstroom is de logische volgorde van activiteiten waarin iets (een project bijvoorbeeld) wordt uitgevoerd.
<b>Widget</b>	Een <i>widget</i> is een klein venster of andere toevoeging die aan een groter geheel kan worden toegevoegd en een specifieke functie heeft.
<b>Gantt chart</b>	Een <i>Gantt chart</i> is een staafdiagram dat de structuur en tijdsplanning van een project weergeeft. Het wordt gebruikt om projecten te monitoren.
<b>Burndown chart</b>	Een <i>burndown chart</i> is een grafische representatie van het werk dat er nog moet uitgevoerd worden en de verlopen tijd.
<b>Access management</b>	<i>Access management</i> omvat het geheel van processen gerelateerd aan authenticatie en autorisatie.
<b>Git repository</b>	Een <i>git repository</i> is een gemeenschappelijke online opslagplaats met ingebouwde versiebeheer bedoeld voor de opslag van softwarecode.
<b>Plug-in</b>	Een <i>plug-in</i> of invoegtoepassing is een uitbreiding op een computerprogramma.
<b>Bottleneck</b>	Een <i>bottleneck</i> of knelpuntfactor is een plek binnen een systeem waarvan de werking ervan heel bepalend is voor de rest van het systeem. Dit kan een proces zijn maar het kan ook een persoon zijn binnen een team.
<b>(Game) engine</b>	Een <i>engine</i> is een softwarematige basis van een computerprogramma. Het wordt vaak gebruikt als tool binnen gamedevelopment.
<b>Automated testen</b>	<i>Automated testen</i> is een andere verwoording voor automatische testen.
<b>Scope</b>	Een <i>scope</i> is een ander woord voor het bereik of de omvang van bijvoorbeeld een opdracht of project.
<b>Testing framework</b>	Een <i>testing framework</i> is een tool die gebruikt wordt bij het testen van software.

## Lijst van gebruikte afkortingen

Afkorting	Verklaring
<b>SRS</b>	Een SRS document of <i>Software Requirements Specification</i> is een communicatie tool tussen stakeholders en softwareontwikkelaars. Het omvat functionele en niet-functionele vereisten voor de ontwikkeling van een softwareproduct. De informatie in het document kan worden ondersteund door <i>use cases</i> en <i>mock-ups</i> .
<b>IDE</b>	Een IDE of <i>integrated development environment</i> is software die een softwareontwikkelaar ondersteunt bij het ontwikkelen van software. Zo een IDE omvat bijvoorbeeld een <i>editor</i> om ontwikkelde software te schrijven of te wijzigen, een <i>compiler</i> om de software te controleren en uitvoerbaar te maken en een <i>debugger</i> om fouten in software op te sporen en te verhelpen.
<b>QA</b>	QA of <i>quality assurance</i> gaat over de activiteiten die uitgevoerd worden om de kwaliteit van een product te waarborgen.
<b>VR</b>	VR of <i>virtual reality</i> is een term die gebruikt wordt om computer gesimuleerde realiteit te benoemen.
<b>FTT</b>	Een FTT of een <i>feature to test</i> is een eigenschap van ontwikkelde software waarvoor een test moet worden gemaakt.
<b>B2B</b>	Bij B2B ook wel <i>business to business</i> genoemd, spreken we over handel specifiek tussen verschillende bedrijven.
<b>B2C</b>	B2C ofwel <i>business to consumer</i> gaat over handel tussen bedrijven en consumenten.
<b>CSV</b>	Een CSV bestand ofwel een <i>comma-separated values</i> bestand, ook bekend als kommagescheiden bestand, is een bestand waarin waarden opgeslagen worden gescheiden met een komma. De data is op die manier gestructureerd alsof het opgeslagen is in een tabel.
<b>HTML</b>	HTML of <i>hypertext markup language</i> is een documenttype dat gebruikt wordt voor het opmaken van documenten op het internet.
<b>ESRB</b>	De <i>Entertainment Software Rating Board (ESRB)</i> is een onafhankelijke organisatie die bepaalt voor welke leeftijden inhoud van bijvoorbeeld games, geschikt zijn.
<b>PEGI</b>	<i>Pan European Game Information</i> of PEGI is een Europees leeftijdsclassificatiesysteem.
<b>API</b>	Een <i>application programming interface</i> of API is een verzameling definities op basis waarvan een computerprogramma kan communiceren met een ander programma of onderdeel (meestal in de vorm van bibliotheken).
<b>SAAS</b>	SAAS of <i>Software as a Service</i> is software die als een online dienst wordt aangeboden.

## Inleiding

Binnen traditionele softwareontwikkeling zijn er vaak meerdere rollen binnen een bedrijf. Werknemers focussen zich op taken zoals softwaretesten, projectmanagement, agile softwareontwikkelingsmethodes, ... om de kwaliteit van het eindproduct te kunnen garanderen. Ook staan deze mensen er voor in dat er exact afgeleverd wordt waar de klant of eindgebruiker om vraagt.

Traditionele software ontwikkeling en gamedevelopment komen op veel vlakken overheen, maar toch verschillen ze van elkaar. Bij beide is er software als eindproduct, maar bij gamedevelopment zijn de teams heel erg multidisciplinair. Artiesten en programmeurs moeten nauw samenwerken, creativiteit en technologie worden vermengt met elkaar en de samenwerking tussen ontwikkelaars en artiesten is niet altijd zo vanzelfsprekend. [1]

Hieruit lijkt het logisch dat zeker binnen gamedevelopment dezelfde noden zouden moeten bestaan zoals binnen traditionele software ontwikkeling, zoals het opvolgen en managen van projecten en het testen van de games om hun kwaliteit te kunnen garanderen. In dit eindwerk is daarom onderzoek gedaan naar projectmanagement binnen gamedevelopment en ook hoe er binnen gamedevelopment aan test kan worden gedaan.

In dit document worden de noden op vlak van software management van een kleine gamedevelopment studio, in kaart gebracht en nader onderzocht. Dit om meer duidelijkheid te scheppen op dit onderwerp.

# I. Stageverslag

## 1. Bedrijfsvoorstelling

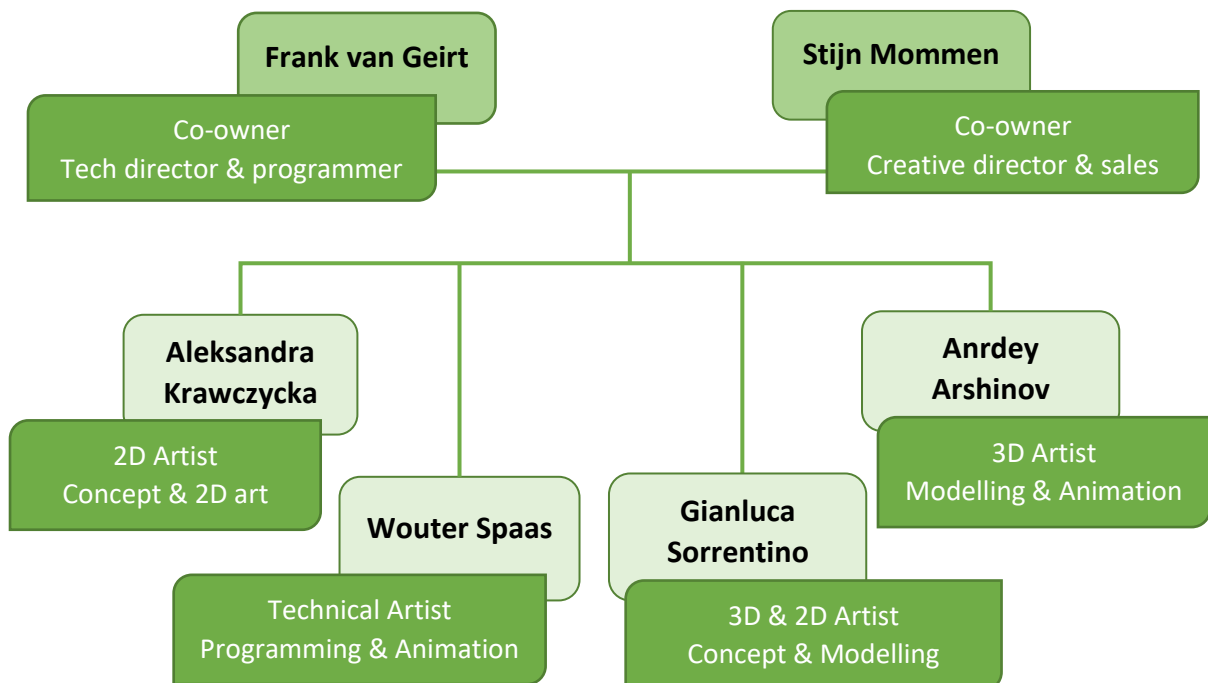
### 1.1 Algemene bedrijfsgegevens

Firewolf Studios is een game- en animatiestudio. Ze focussen zich op het ontwikkelen van serious games. Deze serious games richten zich op het trainen van werknemers, verzamelen van leads, informeren van prospects of informeren van klanten. Belangrijk voor hun is dat zo een serious game wel steeds 'fun' is. Ze zijn op die manier dus uniek in het feit dat ze enkel focussen op games met een serieus doel in de plaats van de vaker voorkomende games gericht op entertainment, maar zorgen er toch voor dat hun games naast het leveren van informatie fijn zijn om te spelen.

De activiteiten binnen het bedrijf gaan van idee- en conceptontwikkeling, over het uitwerken van prototypes tot het volledig ontwikkelen en beheren van afgewerkte games. Ze bevinden zich dus ook in de IT sector, maar focussen zich eerder op het creatieve domein hierin. Ze gebruiken de IT business om informatie te brengen naar gebruikers op een creatieve en speelse manier.

### 1.2 Bedrijfsstructuur

Firewolf Studios is een kleine gamedevelopment studio bestaande uit zes leden met ieder een specifieke rol binnen het team. Toen de stageperiode begon bestond het team nog uit vijf leden, maar tijdens de stageperiode is er een teamlid bijgekomen. Figuur 1 toont een organogram van de structuur van het stagebedrijf. Bovenaan staan de twee bedrijfsleiders, en onderdaan de vier werknemers.



Figuur 1: Organogram van de bedrijfsstructuur van Firewolf Studios

### 1.3 Klanten en projecten

De studio bestaat sinds 2012 en heeft al een hele reeks projecten mogen afwerken in die tijd in verschillende sectoren.

Zo zijn ze bijvoorbeeld al in aanmerking gekomen met de gezondheidszorg als doelgroep voor hun projecten. De game *Autiworld* is daar een voorbeeld van. Deze game laat de speler toe een kijkje te nemen in de wereld van iemand met autisme. Het is een 3D spel, gespeeld vanuit de eerste persoon waarin je alledaagse taken uitvoert. Tijdens het uitvoeren van die taken kom je in aanraking met verscheidene problemen waar mensen die autisme hebben dagelijks mee kampen. Het is een publiek toegankelijk spel wat iedereen gratis online kan downloaden.

*Be Ready* is een ander voorbeeld van een game ontwikkeld door Firewolf Studios. Deze game is ontwikkeld naar burgemeesters toe om hen te trainen in het maken van de juiste beslissingen in noodsituaties.

*Greenracer* is dan weer een intern project vanuit Firewolf Studios. Dit is een mobiele 3D game die ook in VR gespeeld kan worden. Hierbij gaat het dan wel om VR voor smartphones. Het is een racespel waarin er geracet wordt met een wagen die stofzuigt. In het spel kan er tijdens de race afval opgezogen worden om zo sneller te kunnen gaan in de race en makkelijker te kunnen winnen.

De game waar verder in het document de allereerste keer mee getest is door middel van *automated* scripts, is *Voltaire*. Dit spel gaat over een kerkhaan met de naam Voltaire die maar 1 doel heeft: de top van de kerktoren bereiken. Het is een 'tower defence' game, dit is een soort game waar vijanden in golven de speler aanvallen. De speler kan zich verdedigen door allerlei verdedigingsmechanismen op het veld te plaatsen en wint wanneer hij de aanvallers succesvol kan verslaan voor de vijand dat doet.

Buiten games, werkt de studio momenteel ook aan een serre configuratiesysteem. In dit programma kunnen gebruikers afmetingen en andere specificaties ingeven voor een serre en laat het programma vervolgens in 3D zien hoe dat er uit zou zien.



## 2. Voorstelling stageopdracht

### 2.1 Probleemstelling

Het eerste verbeterpunt binnen het stagebedrijf is het overzicht op de huidige projecten. Er is geen duidelijk zichtbaar overzicht van de huidige vooruitgang binnen de projecten of van de nog af te werken taken voor de lopende projecten. De teamleden weten allemaal hoe ver ieder project ervoor staat, maar buiten papieren checklists wordt deze informatie nergens bijgehouden. Er is geen duidelijk visuele voorstelling van de voortgang van een project en het is moeilijk om in te schatten hoe veel tijd nieuwe projecten of taken in beslag zullen nemen of hoeveel tijd er nog effectief nodig is voor een project afgerond kan worden.

Binnen de testprocedure van het bedrijf zit een tweede verbeterpunt. Er wordt niet uitgebreid getest waardoor de kwaliteit van de eindproducten niet diepgaand kan worden ondersteund.

### 2.2 Doelstelling

Om de werking van het bedrijf te verbeteren op vlak van projectmanagement, worden verschillende agile methodologieën toegepast binnen het bedrijf om hier meer inzicht in te verkrijgen. Ook wordt er onderzocht of dit vaker toegepast wordt binnen gamedevelopment of net niet in een literatuurstudie. Dit maakt deel uit van het onderzoeksgedeelte. Hierbij worden verschillende agile methodes onderzocht, maar ook andere methodologieën worden besproken. Ook is hieruit een onderzoeksvraag ontstaan. Er is onderzoek gebeurd naar de toepassing van issuetracking- en projectmanagement tools binnen de studio.

Om de testprocedure te verbeteren wordt er gezocht naar testmethodes binnen gamedevelopment. Ook wordt er geanalyseerd welke testmethodes het bedrijf kan toepassen om hun testprocedure te verbeteren op een manier die bij hun werkwijze past. Er is onderzocht op welke manieren men testen kan uitvoeren binnen de ontwikkelingsomgeving naar keuze van het bedrijf: Unity. Er zijn testen uitgevoerd met Unium en Jasmine, Python en PyAutoGUI en de mogelijkheid van performancetesten is bekeken. Ook is er bekeken hoe het bedrijf de informatie uit deze testen kan documenteren.

## 3. Uitwerking stageopdracht

### 3.1 Beschrijving aanpak

Om de stageopdracht grondig uit te voeren is er steeds eerst het nodige opzoekingswerk gedaan vooraleer er gestart werd met een implementatie. Dit ging vaak over het lezen van artikels, maar ook veel informatie en ideeën zijn gevolgd uit de literatuurstudie die terug te vinden is in het tweede deel van dit document: het onderzoek. Indien er iets geïmplementeerd wordt, worden de resultaten hiervan waar mogelijk ook vergeleken met die van andere bronnen. Oplossingen die uiteindelijk niet geïmplementeerd zijn maar wel onderzocht zijn, worden ook vermeld en besproken.

### 3.2 Softwaretesten binnen gamedevelopment

#### 3.2.1 Vergelijking gametesten en traditionele softwaretesten

Alvorens de testsituatie van Firewolf Studios nader te bekijken of nieuwe testmethodes te implementeren binnen het bedrijf, wordt er eerst gekeken naar de mogelijkheden van testen binnen gamedevelopment en wordt er onderzocht wat de verschillen zijn met traditionele softwaretesten.

Traditionele softwareontwikkeling en het ontwikkelen van games verschillen van elkaar. Bij beide wordt er software ontwikkeld, maar bij gamedevelopment is er een samenwerking van mensen uit zeer verschillende disciplines. In een gamedevelopmentomgeving is bijvoorbeeld niet altijd iedereen binnen het team technisch van achtergrond. Ook verschillende soorten artiesten kennen hun plek in een team dat games ontwikkelt.

Het testen van software en het testen van games wordt beide gedaan omwille van hetzelfde doel, maar toch verschillen ze van elkaar. Het testen van software is een proces waar de kwaliteit van software onderzocht wordt door testen uit te voeren: zo kunnen er *bugs* en andere defecten uit de gemaakte software gehaald worden en kan er geverifieerd worden of het product werkt zoals verwacht. Het testen van games is ook een testproces zoals dat bij gewone software het geval is, maar dan specifiek gericht op de kwaliteitscontrole van games. Games zijn ook niet alleen opgebouwd uit logische functionaliteiten, maar bevatten ook kunst in de vorm van bijvoorbeeld 3D-modellen die moeilijker te testen zijn. Ook testen hoe 'fijn' een game is, kam moeilijk beschreven worden in een testcase en kan enkel blijken uit de ervaring van de game te spelen.

In het volgende onderdeel worden enkele vormen van testen die typisch binnen gamedevelopment voorkomen en toegepast worden, nader besproken.

### 3.2.2 Gekende testmethodes binnen gamedevelopment

Tabel 1 toont enkele vormen van testen die voorkomen binnen gamedevelopment. [2]

Testmethode	Verklaring
<b>Conformiteitstesten</b>	<p>Voor het ontwikkelen van games voor bepaalde consoles (Playstation van Sony, Xbox van Microsoft), ... moeten richtlijnen gevolgd worden die opgesteld zijn door de fabrikant. Het gaat vaak over hoe foutberichten binnen de code behandeld worden, het behandelen van data in memory cards, het behandelen van bestanden waar copyright op rust en dergelijke. Om een licentie te krijgen om een game te mogen uitbrengen voor een bepaald platform, wordt er geverifieerd of deze richtlijnen wel nageleefd zijn. Testers voeren daarom testen uit om het navolgen van deze richtlijnen te controleren alvorens er een licentie aangevraagd wordt.</p> <p>Conformiteitstesten kan ook gaan over de leeftijdscategorie die bepaald wordt voor een game. Dit wordt gedaan volgens richtlijnen van bijvoorbeeld ESRB of PEGI. De testers zijn verantwoordelijk voor het bepalen of de inhoud van het spel past bij de leeftijd van de doelgroep.</p>
<b>Compatibiliteitstesten</b>	<p>Binnen compatibiliteitstesten wordt de prestatie van een game geanalyseerd op verschillende systeemconfiguraties. Op deze manier kan bepaald worden wat de minimumsysteemvereisten zijn voor het systeem van de eindgebruiker.</p>
<b>Lokalisatietesten</b>	<p>Vaak is een game in meerdere talen beschikbaar. Met het uitvoeren van lokalisatietesten wordt het controleren van de taal in een game bedoeld. Er wordt dus gecontroleerd of de game steeds correct vertaald is.</p>
<b>Soaktesten</b>	<p>Bij soaktesten wordt een game gedurende een hele lange tijd uitgevoerd, vaak gepauzeerd of in een menuscherm. Dit is een manier om fouten zoals een geheugenlek en fouten op vlak van het afronden van getallen bloot te leggen, aangezien het effect hiervan pas zichtbaar wordt als het programma gedurende een langere tijd wordt uitgevoerd.</p>
<b>Regressietesten</b>	<p>Regressietesten worden uitgevoerd nadat een ontwikkelaar een aanpassing heeft gemaakt aan de game. Dit kan gaan over het toevoegen van een <i>feature</i> of het oplossen van een <i>bug</i>. Door middel van een regressietest wordt er gecontroleerd of de nieuwe toegevoegde <i>feature</i> correct werkt, of de hele game daardoor nog correct werkt of er wordt gecontroleerd of een <i>bug</i> door middel van de aanpassingen opgelost is.</p>
<b>Loadtesten</b>	<p>Via loadtesten worden de limieten van een game getest. Dit kan gaan over het aantal elementen die tegelijk in beeld kunnen zijn voor een game traag wordt, of het kan gaan over het aantal gebruikers dat de game tegelijkertijd kan spelen.</p>

Tabel 1: Gekende testmethodes binnen gamedevelopment

## 3.2.3 Gamedevelopment en testen binnen Firewolf Studios

### 3.2.3.1 Doelplatvormen van Firewolf Studios

Om nieuwe testmethodes te kunnen introduceren binnen het bedrijf, is het belangrijk om te weten wat voor software er ontwikkeld wordt en dus ook getest wordt. Binnen het bedrijf worden alle games ontwikkeld met de *game-engine* Unity. De doelplatformen waarvoor het bedrijf met deze *engine* games produceert zijn:

- WebGL (games in de webbrowser)
- Android en iOS (smartphones)
- Windows (pc's)

### 3.2.3.2 Huidige testsituatie

Ook wordt de huidige testsituatie geobserveerd en in kaart gebracht. Op deze manier kunnen verbeterpunten gevonden worden. Ze worden niet bij naam genoemd, maar er worden wel al enkele vormen van testen gebruikt binnen de studio. Deze worden nader beschreven in tabel 2.

<i>Soort test</i>	<i>Verklaring</i>
<b>Compatibiliteitstesten</b>	De ontwikkelde games worden gespeeld op verschillende toestellen. Dit vaak uit noodzaak omdat klanten specifiek vraag hebben naar het ontwikkelen van games op bepaalde platformen. Het bedrijf is op die manier genoodzaakt hun games te optimaliseren voor ieder platform waar de klant of het doelpubliek om vraagt. De games worden getest op een iPad Pro, een iPhone, een paar Android toestellen van de werknemers (enkele wat meer high end en enkele wat meer low end) en een medium-end desktop.
<b>Acceptatietesten</b>	Bij een project dat op dit moment loopt ontvangt de klant van zijn game van tijd tot tijd een versie via mail. Er wordt verwacht van de klant dat er feedback op komt, zodat het project eventueel bijgestuurd kan worden.
<b>Gametesten</b>	Natuurlijk worden de games in de studio ook in hun geheel getest, dit vaak door een game gewoon te spelen. Dit kan gebeuren tijdens het ontwikkelen, na het afwerken van een onderdeel of wanneer de game volledig ontwikkeld is. Door telkens te spelen na het ontwikkelen van een nieuw onderdeel wordt er ook al een beetje aan regressietesten gedaan. Het spel spelen na het toevoegen van een nieuw onderdeel kan ervoor zorgen dat men snel opmerkt of dit nieuwe onderdeel een ouder onderdeel doet falen.

Tabel 2: Soorten testen binnen Firewolf Studios

Hierbij komt dat er binnen het bedrijf geen aangewezen persoon is die de games test of de kwaliteit ervan garandeert. Wanneer er nieuwe projecten gepland worden, worden er altijd wel dagen voorzien om testen uit te voeren maar hiervoor is er geen vaste procedure aanwezig. Deze situatie kan verbeterd worden door meer verschillende testmethodes toe te passen, wat de kwaliteit van de games binnen het bedrijf beter kan garanderen. De uitwerking hiervan volgt in deel 3.2.4

### 3.2.3.3 Bespreking toepassing van gekende testmethodes binnen gamedevelopment

In dit onderdeel worden de testmethodes die gekend zijn binnen gamedevelopment en eerder besproken zijn in deel 3.2.2, besproken in het kader van het stagebedrijf. Zie hiervoor tabel 3.

<i>Testmethode</i>	<i>Bespreking</i>
<b>Conformiteitstesten</b>	Aangezien Firewolf Studios vooral B2B werkt en de meeste van hun games niet publiek zijn maar ontwikkeld zijn voor een bedrijf, en aangezien ze ook geen games ontwikkelen waar consoles het doelplatform zijn, komen ze voorlopig niet in aanraking met dit soort testen.
<b>Lokalisatietesten</b>	Lokalisatietesten komen momenteel niet voor binnen Firewolf Studios en het lijkt er niet op dat men hier in de nabije toekomst rekening mee zal moeten houden. Het bedrijf werkt voorlopig nog niet internationaal, waardoor al hun games in het Nederlands ontwikkeld worden en vertalingen niet nodig zijn.
<b>Soak testen</b>	Soak testen worden niet toegepast maar zouden geïmplementeerd kunnen worden. Het vereist wel dat een game continue kan blijven draaien op een toestel dat niet uitgeschakeld wordt. Ook zouden er dan loggegevens van het spel moeten bijgehouden worden om effectief te kunnen gaan analyseren of er problemen optreden.
<b>Regressietesten</b>	Regressietesten worden reeds informeel uitgevoerd. Men wacht niet tot het einde van de ontwikkeling om een game al eens uit te voeren. Dit kan verbeterd worden door de games te spelen op specifieke momenten binnen het development proces. Zo kunnen problemen veroorzaakt door een nieuw onderdeel vroegtijdig opgespoord worden.
<b>Loadtesten</b>	Momenteel is er geen enkele game binnen de studio waar men een groot aantal concurrente gebruikers zal hebben. Ook zijn er geen games in ontwikkeling waar de gebruiker invloed heeft op hoeveel elementen er bijvoorbeeld tegelijk in beeld voorkomen. Het uitvoeren van loadtesten is daarom nog niet noodzakelijk voor het bedrijf. Dit wordt noodzakelijk van zodra ze beginnen met het ontwikkelen van online games waarin meerdere gebruikers tegelijk kunnen spelen.

*Tabel 3: Bespreking toepassing van gekende testmethodes binnen gamedevelopment*

### 3.2.4 Nieuwe vormen van testen binnen Firewolf Studios

Om de testprocedure binnen Firewolf Studios te verbeteren werd er tijdens de stageperiode met enkele nieuwe vormen van testen geëxperimenteerd binnen het bedrijf, om vervolgens te kunnen besluiten of ze nuttig kunnen zijn voor hun. Indien hun nut zich bewijst kunnen ze geïntroduceerd worden binnen het bedrijf. Dit zijn geautomatiseerde testen met Unium en JavaScript, Python en PyAutoGUI en performancetesten. Ze worden uitgebreid beschreven in de volgende hoofdstukken.

Het bedrijf heeft nog geen uitgebreide testprocedure, maar toch zijn er artikels en professionals die ervoor pleiten om een uitgewerkte testprocedure te hanteren binnen gamedevelopment. In een artikel geschreven door twee professionals in de game industrie, wordt dit namelijk uitgebreid beargumenteerd. [3] Er wordt geschreven dat belangrijke kwaliteiten van goede software de volgende zijn: compleetheid, correctheid, goede prestaties en stabiliteit. Doet de software alles wat het moet doen? Doet het dat zonder errors? Werkt het programma onder acceptabele prestaties? Handelt het programma onverwachte fouten goed af? En als bonus factor binnen gamedevelopment komt daar nog eens bij dat een game ook fijn moet zijn om te spelen.

Het artikel haalt aan dat een testproces of procedure ontwikkelen de eerste stap is naar goede testen uitvoeren. Dit is een proces dat veel tijd in beslag neemt en waarbij er continue op zoek moet worden gegaan naar verbeteringen. Binnen de stageopdracht wordt er bekeken welke manieren van testen toegepast kunnen worden op de studio, om zo een eerste aanzet te geven naar het ontwikkelen van een test procedure.

Het artikel haalt verschillende vormen van testen aan en past ze toe op gamedevelopment. Het legt uit dat unittesten gebruikt worden om delen van code individueel te testen op hun functionaliteit, in een geïsoleerde omgeving. Integratie testen leunt dan weer dicht aan bij unittesten, maar hier worden meerdere functionaliteiten tegelijk getest. Smoke testen worden dan weer gebruikt om het systeem in zijn geheel te testen.

Het artikel raadt aan om verder te gaan dan gewoon scripts te schrijven om *automated testen* mogelijk te maken. Het raadt aan om ze zeker ook op een *continuous integration* systeem te plaatsen. Op deze manier weet het development team snel of een nieuwe *feature* een verkeerd effect heeft op de rest van de game. De auteurs van het artikel gebruiken hier Jenkins voor als tool. Het implementeren van zo een systeem valt buiten de *scope* van de stageopdracht, maar dit wordt wel als uitgangspunt gezien bij het zoeken naar manieren voor het stagebedrijf waarop het aan *automated testen* kan gaan doen.

Een ander belangrijk element dat de auteurs aanraden is het gebruik van een issuetrackingsysteem. Het gebruik hiervan, toegepast op het stagebedrijf, wordt tijdens het onderzoeksgedeelte binnen dit eindwerk nader bekeken.

Het testen van de prestaties van een game is ook belangrijk, er wordt aangeraden om data op vlak van prestaties altijd te loggen gedurende de ontwikkeling van het spel. De data kan op die manier altijd vergeleken worden met die van een vorige versie om zo problemen tijdig op te merken en te kunnen analyseren.

Met de kennis uit dit artikel zijn volgende topics op vlak van testen binnen gamedevelopment verder onderzocht.

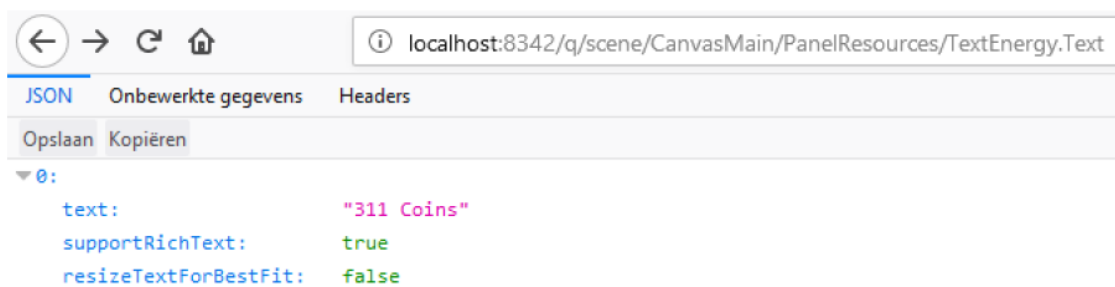
### 3.2.4.1 Geautomatiseerd testen met Unium

Tijdens de stageperiode is er onderzoek gedaan naar een mogelijkheid om het testen van projecten gemaakt binnen Unity automatisch te maken.

Dit onderzoek is gestart na het ontdekken van een asset binnen de assetstore van Unity met de naam 'Unium'. Unium is een *library* en tool om ondersteuning te bieden voor geautomatiseerde testen in Unity. Unium doet dit door:

- Een web server te implementeren in het project om een *interface* te bieden aan de game
- Een query taal aan te leveren om makkelijker via deze *interface* data uit het project op te vragen

Als eerste stap moet Unium geïmporteerd worden in een Unity project naar keuze. Bij de bestanden die geïmporteerd worden zitten een script dat moet verbonden worden aan een leeg GameObject in de hoofdsceen van het project. Hierna kan er data uit het project opgevraagd worden via een webserver door gebruikt te maken van de query taal die door Unium is geïmplementeerd. Een voorbeeld hiervan is de zien in figuur 2. De data die terug wordt gegeven wordt getoond in JSON formaat.



Figuur 2: Voorbeeldquery in Unium en resulterende JSON data

Het opzetten van een manier om data uit het project te verkrijgen is de eerste stap, hierna moeten er scripts ontwikkeld worden die testen of deze JSON objecten wel de data bevatten die we verwachten.

De keuze van een *testing framework* is volledig vrij omdat Unium een API aanbiedt dat eender welk framework kan aanroepen. Er is gekozen om de testen te schrijven in JavaScript met behulp van het Jasmine framework (versie 2.8.0). Jasmine is een JavaScript *testing framework*.

De code wordt gerund via Node.js (versie 9.2.1) als platform. Hier worden de testen niet uitgevoerd via de webbrowser maar in een JavaScript *engine* van Node.js zelf.

In figuur 3 volgt een voorbeeld van zo een script dat op deze manier is ontwikkeld. Het maakt gebruik van de query die terug is te vinden in figuur 2 in de adresbalk. De test case omvat het testen of de game het correcte aantal muntjes aan de speler geeft tijdens het uitvoeren van de game.

```

const request = require("request").defaults({json:true});

describe("CanvasMain", () => {

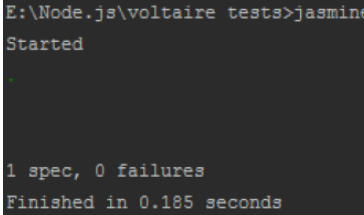
  it("claims that the game starts with 27 coins and adds 2 coins every second", (done) => {
    request('http://localhost:8342/q/scene/Timer.TimerTest', (err, response, body) => {
      request('http://localhost:8342/q/scene/CanvasMain/PanelResources/TextEnergy.Text', (err2, response2, body2) => {
        expect(parseFloat(body2.text)).toEqual(27+(body.timer*2));
        done();
      });
    });
  });
});

```

Figuur 3: Voorbeeldscript gebruikmakend van Unium en Jasmine

Om de huidige tijd in de game op te kunnen vragen is er een simpel C# script geschreven dat de huidige tijd teruggeeft. Dit wordt aangeroepen in de eerste query. Deze eerste query volgt op de vierde regel, na het woord 'request'. Het huidige aantal muntjes wordt opgevraagd door de tweede query op de vijfde regel. Er is geweten dat de relatie tussen de tijd en muntjes in deze game als volgt is: er komen 2 muntjes bij per seconden en er wordt gestart met 27 muntjes. Vandaar de formule:  $27 + (\text{verstreken tijd} * 2)$ . Door middel van 'expect' te gebruiken wordt er getest of het aantal muntjes wel degelijk overeenkomt met het aantal muntjes dat we verwachten.

Om de test ook effectief te runnen, wordt het 'jasmine' commando uitgevoerd in de terminal. Figuur 4 toont het resultaat daarvan.



```

E:\Node.js\voltaire tests>jasmine
Started

1 spec, 0 failures
Finished in 0.185 seconds

```

Figuur 4: Voorbeeld van de uitvoering van een test in Jasmine

Jasmine heeft 1 *spec* gevonden in de *spec* map van het project en heeft deze bijgevolg uitgevoerd. Een *spec* is binnen Jasmine een andere naam voor een test. Hierbij kwamen geen *failures*, voor wat dus betekent dat de test het verwachte resultaat teruggaf en de test geslaagd is.

Het schrijven en uitvoeren van deze testen wordt gezien als *automated testen*. Wanneer de code van het project een aanpassing is ondergaan, kan jasmine opnieuw de testen uitvoeren en tonen of de testen nog steeds slagen. Jasmine voert de testen uit, ze hoeven niet manueel meer uitgevoerd te worden.

De volgende stap hierin zou *continuous integration* zijn: hiervoor hebben is er een automation server nodig zoals bijvoorbeeld een Linux server met Jenkins die telkens wanneer er een nieuwe commit is op bijvoorbeeld *Git*, het Unity project buildt en er de testen op uitvoert en de resultaten ervan logt. Dit valt buiten de *scope* van de stageopdracht.



De voor- en nadelen van testen van games met Unium op een rijtje in tabel 4.

Voordelen	Nadelen
Mogelijkheden tot <i>continuous integration</i>	Testen zijn gelimiteerd: er kunnen geen methodes opgeroepen worden en daardoor ook geen nieuwe objecten geïnstantieerd worden.

Tabel 4: Voor- en nadelen van Unium

### 3.2.3.2 Geautomatiseerd testen met Python

Een andere manier waarop een game getest kan worden op een geautomatiseerde manier, is door middel van scripts met Python. Door middel van de PyAutoGUI module is het mogelijk om de input van muis en toetsenbord te automatiseren. Dit laat toe een game herhaaldelijk af te spelen terwijl er muis- en toetsenbordhandelingen gesimuleerd worden. De game moet dus niet meer manueel gespeeld worden en handelingen moeten niet meer manueel herhaald worden om de game te testen. Dit zorgt voor tijdsbesparing bij het testen: het script wordt één keer geschreven en kan daarna herhaaldelijk opnieuw worden uitgevoerd.

Een voorbeeld uitgewerkt als onderdeel van de stageopdracht werkt als volgt: er is een python script geschreven dat elke twee seconden de huidige coördinaten van de muis print. Dit script (figuur 5) wordt als een soort tool gebruikt om te ontdekken op welke coördinaten knoppen en dergelijke in het spel zich bevinden. In een ander script (figuur 6) kunnen deze coördinaten dan gebruikt worden om effectief handelingen (testen) uit te voeren.

```
#!/ python3
# voltaire tool.py - Displays current mouse position every 2 seconds
import time, pyautogui

try:
    while True:
        x, y = pyautogui.position()
        position = 'X: ' + str(x) + ' Y: ' + str(y)
        print(position, '\n')
        time.sleep(2)
except KeyboardInterrupt:
    print('\nDone.')
```

Figuur 5: PyAutoGUI tool voor Voltaire

```
#!/ python3
# voltaire mudpool.py - Place a mudpool on Hotspot 1 in Voltaire
import pyautogui
pyautogui.PAUSE = 2

pyautogui.click()
pyautogui.moveTo(56, 639)
message = 'Positioned over button mudpool: '
print(message + str(pyautogui.position()))
pyautogui.dragTo(369, 896, duration=0.2)
```

Figuur 6: Voorbeeld van een test voor Voltaire met PyAutoGUI

Er kruipt wel tijd in het verzamelen van de coördinaten en deze vervolgens in een script te plaatsen, wat gezien kan worden als een nadeel. Een ander nadeel is dat deze manier van testen niet zou mogen werken in een afgewerkte game. Het zou betekenen dat de eindgebruiker het spel kan *botten*. Bij een game waar men bijvoorbeeld vaak een handeling moet herhalen in een spel en waarbij die herhaling kan worden vermeden door een aankoop te doen in de game, zou deze aankoop vermeden kunnen worden vermeden door die handeling automatisch te maken op deze manier. Indien gebruikers de game dus automatiseren in de plaats van aankopen te doen, veroorzaakt dit een daling in de inkomsten voor de ontwikkelaars van de game.

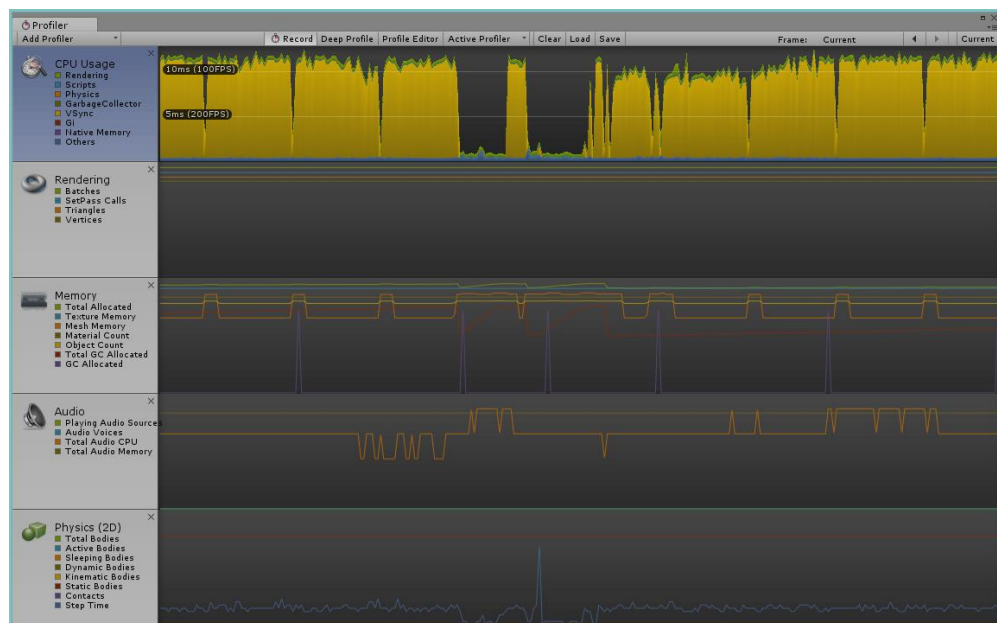
De voor- en nadelen van het testen van games met PyAutoGUI op een rijtje in tabel 5:

Voordelen	Nadelen
Tijdbesparing: het spel moet niet meer manueel gespeeld worden om het te testen.	Dit zou niet mogen werken in een afgewerkte game, het bevestigt dat er mogelijkheden zijn tot het 'botten' van de game.
	Hoewel het voor tijdbesparing kan zorgen, kan er ook veel tijd kruipen in het maken van de scripts.

Tabel 5: Voor- en nadelen van PyAutoGUI

### 3.2.3.3 Performancetesten

Om de prestaties van een spel te kunnen testen en analyseren op bijvoorbeeld verschillende systemen, wordt er aan prestatietesten gedaan. Unity heeft hier een eigen oplossing voor, de 'profiler'. Dit is een scherm in de editor die aangezet kan worden en vervolgens tijdens de game prestatiedata gaat opnemen. De profiler toont deze data op een tijdslijn. Op die manier kan er gezien worden waar er nog knelpunten zitten in de game die er bijvoorbeeld voor zorgen dat het systeem meer resources gebruikt dan verwacht. [4]

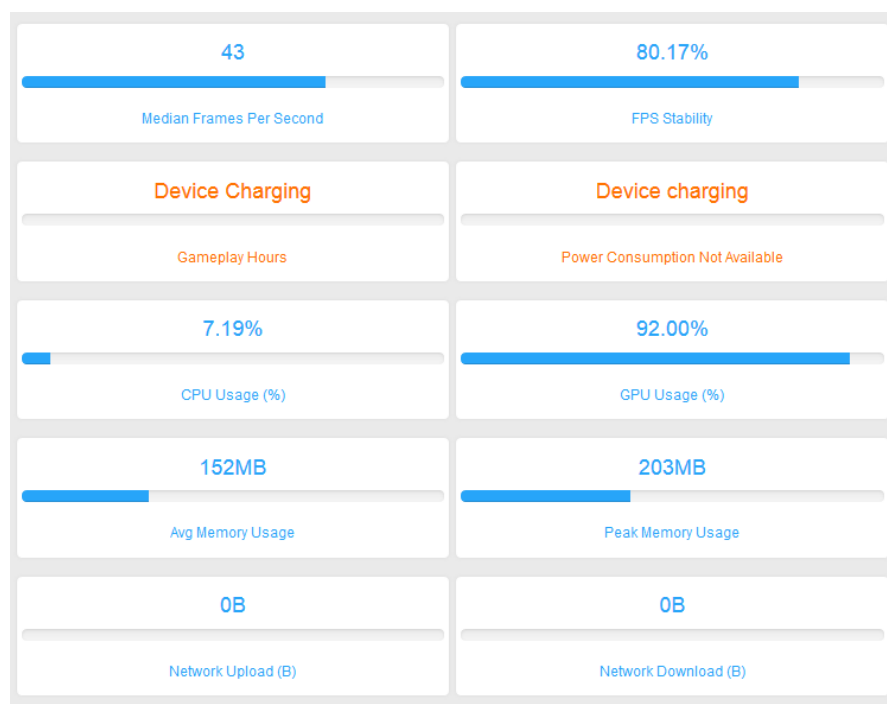


Figuur 7: Unity Profiler

Figuur 7 toont hoe de Unity profiler eruit ziet tijdens de werking van een game uit Firewolf Studios. Unity weergeeft data over verschillende categorieën gaande van de belasting op de CPU, het geheugen en het netwerk. Niet alle categorieën pasten binnen de figuur.

De games kunnen natuurlijk ook getest worden op hun prestaties door middel van een externe tool. Zo is er de tool 'GameBench' gebruikt om games te monitoren wanneer ze op een smartphone werden gespeeld. Op die manier kan er gezien worden hoe een game effectief speelt op een smartphone met bepaalde hardware specificaties. De tool laat de gebruiker toe een spel op zijn device naar keuze te spelen, terwijl het de prestaties van het spel logt op een verbonden PC. De prestatiemeting gaat onder handen over het gebruik van de CPU, de GPU, mobiele netwerken en het gebruik van de accu.

Om met GameBench te werken wordt er een programma geïnstalleerd op een pc en een app op de smartphone waarop getest moet worden. De smartphone wordt vervolgens verbonden met de sessie op de desktop. Indien via de GameBench app een game wordt gestart op de smartphone, dan worden er op de pc meetwaarden getoond tijdens het spelen van het spel. Deze meetwaarden worden vervolgens online opgeslagen en zijn altijd terug te vinden op de website van het product. Figuur 8 toont een voorbeeld van hoe de meetwaarden van een spel binnen Firewolf Studios eruit zagen. [5]



*Figuur 8: Meetwaarden binnen GameBench*

Unity weergeeft prestatiedata over de game terwijl het zich afspeelt op de hardware van de pc. Stel dat dit een game is met Android als doelplatform, dan kan het dus dat deze resultaten niet accuraat zijn. In tegenstelling tot GameBench, hier worden de games namelijk getest terwijl ze zich afspelen op de doel hardware (een smartphone).

### 3.2.3.4 Documentatie

Het effectieve uitvoeren van testen, het bijhouden van de resultaten van de testen en het analyseren van deze resultaten is een belangrijk onderdeel van testen. Het bijhouden van deze data rond testen kan gebeuren via een testmanagementtool of via documentatie. Voor er bijvoorbeeld scripts geschreven worden om iets te testen in een game, moet hiervoor een testcase worden bedacht.

Om de testprocedure te beginnen ontwikkelen binnen het bedrijf is het ook belangrijk dat er al aan dit onderdeel wordt gedacht. In een eerste instantie lijken de belangrijkste documenten dan documenten te zijn die op lijsten wat de FTT's zijn (een *test approach*). Vervolgens is een document dat exact beschrijft hoe dit getest kan worden nodig (een *test design*). Tijdens de stageperiode werden dit soort documenten ontwikkeld als basis om effectieve testen te schrijven.

In bijlage A van dit document zijn voorbeelden van dit soort documenten, ontwikkeld tijdens de stageperiode, toegevoegd. Een test approach bestand over de game *Voltaire* weergeeft welke eigenschappen van het spel worden getest, wat de prioriteit daarvan is, of de test automatisch verloopt of manueel, of de test al is uitgevoerd en indien ja, wanneer. In de bijlage zijn ook twee voorbeelden van test designs terug te vinden. Deze omvatten steeds een id dat ook voorkomt in de test approach, de eigenschap die getest wordt, de prioriteit en de eigenlijke test samen met het eventuele resultaat ervan en een uitvoerdatum.

## 3.2.5 Niet-geïmplementeerde testmethodes

### 3.2.4.1 Unity test tools

Unity voorziet ook eigen testtools binnen hun product. Zo zit er in de huidige Unity versie (2017.3) standaard een 'Unity Test Runner' ingebouwd. In vorige versies van het product heette dit de 'Editor Test Runner'. Het gebruikt een integratie van de NUnit *library*, wat een open-source unit testing *library* is om .NET talen mee te testen. Deze testtools worden dus gebruikt om *unittesten* uit te voeren. *Unittesten* zijn testen waarbij stukken code afzonderlijk van elkaar worden getest.

Testen kunnen uitgevoerd worden tijdens dat de game nog in *edit* modus is, wat betekent dat de game nog bewerkt wordt. Maar ze kunnen ook uitgevoerd worden tijdens de *play* modus, wat betekent dat testen uitgevoerd kunnen worden terwijl de game afspeelt. [6]

Het gebruik van deze manier van testen hoort te gebeuren door de ontwikkelaars van de software en niet de testers ervan. Hierdoor viel het gebruik van deze testmethode buiten de omvang van de stageopdracht.

## 3.2.6 Resultaten

Een van de doelstellingen van de stageopdracht was het verbeteren van de testprocedure binnen het stagebedrijf. Hiervoor is er onderzoek gedaan naar manieren om games te testen, en meer specifiek het automatiseren ervan. Het bedrijf heeft ondervonden hoe ze kunnen testen in Unity zelf, hoe ze kunnen testen met Unium en Jasmine en hoe ze scripts kunnen schrijven in Python met PyAutoGUI om muis en toetsenbord invoer te kunnen automatiseren. Unium en PyAutoGUI zijn manieren van geautomatiseerd testen. Deze informatie kan hen naar de toekomst toe helpen de kwaliteit van hun producten extra te ondersteunen.

### 3.3 Projectmanagement binnen gamedevelopment

#### 3.3.1 Huidige situatie binnen Firewolf Studios

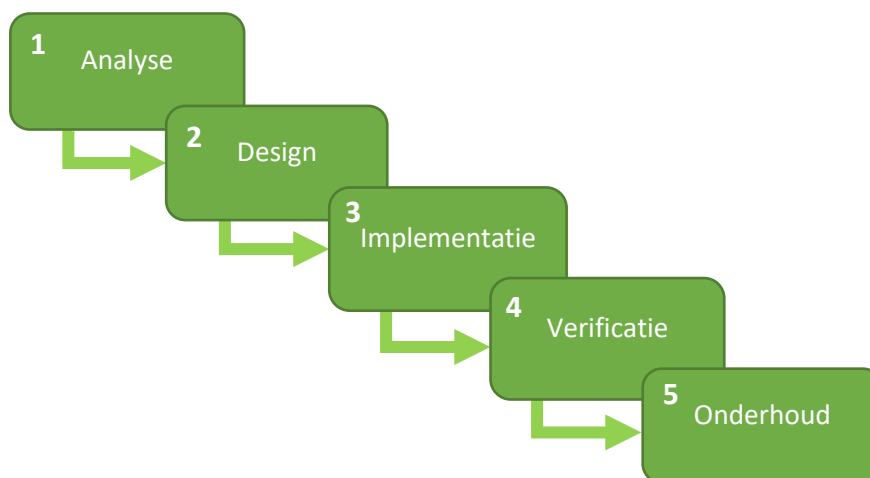
Alvorens de huidige manier van projectmanagement te proberen verbeteren, wordt de huidige situatie nader bekeken.

Er is geen zichtbare vorm van projectmanagement binnen het bedrijf en er worden geen specifieke richtlijnen gevolgd. Er is geen whiteboard of Kanban bord of dergelijke aanwezig op het kantoor om de uit te voeren taken te visualiseren, ook niet online. Uit te voeren taken worden mondeling besproken en verdeeld. De enige vorm waarin een uit te voeren taak soms bijgehouden wordt is een papieren geschreven checklist die als geheugensteuntje dient.

Wel is er een zekere regelmaat in de voortgang van alle projecten die binnen het bedrijf lopen. Er wordt geen *software development model* bij naam genoemd binnen het bedrijf, maar binnen elk project wordt er toch op eenzelfde manier te werk gegaan.

Bij externe projecten naar een klant toe zijn er eerst steeds contactmomenten met de klant waarin er gebrainstormd en overlegd wordt om te bepalen waar de klant naar op zoek is, wat de eisen van de klant zijn. Dit is dus een fase waarin de vereisten van de klant in kaart worden gebracht en er al nagedacht wordt over een mogelijk concept of design. Nadat dit bekend is worden er binnen de studio tekeningen gemaakt om ideeën rond de game te visualiseren en wordt er een specifiek design op papier gezet, het concept wordt intern verder uitgewerkt. Wanneer de klant aangeeft akkoord te zijn met het design wordt er gestart met de ontwikkeling van de game. Na de ontwikkeling kan een klant nog feedback geven en kunnen er eventuele aanpassingen aan de game gemaakt worden. Om dubbel werk te voorkomen en zeker bij grotere projecten last het bedrijf tussentijds ook contactmomenten in met de klant.

De verschillende fases die het bedrijf doorgaans doorloopt binnen een project zouden passen binnen het *waterfall* model. In dit *software development model* bestaan er verschillende fases die sequentieel mekaar opvolgen. De progressie tussen de fases is maar in 1 richting, zo komt de methode ook aan zijn naam. Voorbeelden van fases die gekend zijn binnen deze methode zijn: analyse (verzamelen van requirements), ontwerp of design, implementatie of ontwikkeling, testen en een onderhoudsfase.



Figuur 9: Het waterfall model toegepast op Firewolf Studios

Figuur 9 toont het *waterfall* model visueel. Vervolgens passen we dit toe op het bedrijf om duidelijk te maken hoe dit model bij Firewolf Studios. Zie hiervoor tabel 6.

<i>Fase</i>	<i>Uitleg</i>
<b>Analyse (fase 1)</b>	In deze fase gebeuren de eerste contactmomenten met de klant en wordt er eventueel met de klant gebrainstormd om tot een eerste concept voor een game of andere softwareoplossing te komen.
<b>Design (fase 2)</b>	De ontwerpfase staat voor de fase waarin men intern een eerste design maakt. Er kan nog een contactmoment zijn met de klant om dit ontwerp te verifiëren. In deze fase worden er ook offertes opgesteld.
<b>Implementatie (fase 3)</b>	De implementatie of ontwikkeling staat voor het effectieve ontwikkelen van de game. In deze fase word 3D-modellen, 2D-ontwerpen en software volledig ontwikkeld. Er kunnen nog tussentijds contactmomenten met de klant zijn (via e-mail) om te bevestigen dat er exact ontwikkeld wordt waar de klant naar vraagt.
<b>Verificatie (fase 4)</b>	Vervolgens komt de verificatie of testfase. Het testen gebeurt tijdens de ontwikkeling door de ontwikkelaars maar ook veel op het einde van een project.
<b>Onderhoud (fase 5)</b>	Eigen games die intern zijn ontwikkeld kennen vaak een onderhoudsfase waarin er achteraf nog aanpassing aan de game worden gemaakt indien nodig. Projecten gemaakt voor externe klanten kennen minder vaak een onderhoudsfase.

*Tabel 6: Uitleg waterfall model toegepast op Firewolf Studios*

Het bedrijf werkt al een beetje iteratief en dus ook agile. Door steeds contact met de klant te hebben en hierdoor ontwerpen tijdens de ontwikkeling aan te passen wordt er al flexibel en dus agile gewerkt. De klant krijgt de game of andere softwareoplossing tussentijds te zien en kan er feedback op geven, waardoor er dus iteratief gewerkt wordt en er steeds een beter product ontstaat. Uit deze opvatting is er besloten om agile principes te proberen toepassen ter verbetering van het projectmanagement binnen het bedrijf, als onderdeel van de stageopdracht.

### 3.3.2 Agile binnen gamedevelopment

#### 3.3.2.1 Andere bronnen

In de literatuurstudie, die terug te vinden is in het onderzoeksgedeelte van dit document, wordt er uitgebreid besproken hoe het zit met agile methodes in de gamedevelopment sector.

Uit de literatuur blijkt dat agile zeker zijn opkomst maakt binnen gamedevelopment. Het gebeurt vooral op een informele manier. Veel problemen die in traditionele software ontwikkeling voorkomen, komen ook in gamedevelopment voor. Daarom is het dus niet vreemd dat problemen die binnen de traditionele software ontwikkeling door agile worden opgelost, ook binnen deze sector opgelost kunnen worden door agile methodes te implementeren. [7]

### 3.3.2.2 Kanban en Scrum binnen Firewolf Studios

Kanban en Scrum zijn enkele van de meest populaire methodologieën [8] en ze worden ook vaak samen gebruikt. In dit onderdeel wordt toegelicht waarom net deze methodes zouden passen binnen de studio en haar projecten.

Als we bekijken welke technologie het best past op vlak van de samenstelling van het team, dan zijn beide methoden toepasselijk. Binnen Kanban heeft niemand een specifieke rol binnen het team, is er geen minimum of maximumgrootte van het team en zijn specialisten welkom.

Binnen Scrum zijn de mogelijke rollen in het team: product owner, scrum master en zijn de andere gewoon onderdeel van het development team. Een Scrum team bestaat meestal uit 5 tot 9 teamleden. Vaak gaat het over een *cross-functioneel* team, een team met mensen die ieder een specifieke specialisatie en kennis hebben. Dit past zeker binnen het plaatje van Firewolf Studios: het is een klein team van 6 personen bestaande uit leden die allemaal een eigen specifieke taak hebben binnen het team. Ze zijn gespecialiseerd voor hun specifieke taken zoals 2D artiest, 3D artiest, programmeur of designer.

Kanban wordt meestal toegepast op projecten waar het vooral gaat om onderhoud van software, en minder voor de ontwikkeling van software maar het is wel mogelijk. Scrum wordt dan weer het meest toegepast op de ontwikkeling van nieuwe software, en minder als het gaat om het onderhoud ervan. De projecten binnen Firewolf Studios zijn een soort van mix van beide. Vooral interne projecten kennen een onderhoudsfase, maar ook externe projecten van klanten kunnen deze mogelijkheid hebben.

### 3.3.2.3 Gebruik van andere agile methodologieën binnen gamedevelopment

Kanban en Scrum zijn niet de enige agile methodologieën die gebruikt worden binnen gamedevelopment. De paper *Game Scrum: An Approach to Agile Game Development* door Barbosa en Godoy beschrijft nog twee andere methodologieën. Ze gaan over het toepassen van XP (*extreme programming*) en RUP (*Rational Unified Process*) binnen gamedevelopment. De toepassingen zijn *eXtrem Game Development* of XGD en het *Game Unified Process*.

eXtreme Game Development of XGD is volgens Barbosa en Godoy een toepassing van de agile methodologie XP op gamedevelopment. Hierin wordt bijvoorbeeld multimediamaatstaf ook opgenomen binnen de *continuous integration* van een project, wordt het multimediamaatstaf dus ook getest, wordt het team als een geheel gezien er worden UML schema's gebruikt om de elementen binnen games te beschrijven. Ook moet er gerichte communicatie zijn tussen gamedesigners, ontwikkelaars en artiesten. De methodologie beschrijft hoe XP kan toegepast worden binnen gamedevelopment en raadt hierbij aan wat de visie van het team moet zijn in zijn geheel. Het raadt artiesten aan samen te werken en voor ontwikkelaars raadt het aan om aan *pair programming* te doen.

De Game Unified Process methodologie is volgens Barbosa en Godoy een methodologie die het RUP (Rational Unified Process) model en technieken van XP met elkaar verbindt. RUP is een methodologie die toegepast wordt om veranderingen binnen software te minimaliseren. XP wordt hier vooral aan verbonden vanwege de manieren die het beschrijft om het schrijven van te veel documentatie, tegen te gaan. [1]

De implementaties binnen de stageopdracht zijn beperkt tot Kanban en Scrum, aangezien hier een raakpunt is met de onderzoeksopdracht. De tools die geïmplementeerd zijn als deel van de onderzoeksopdracht, bieden namelijk vooral ondersteuning voor Kanban en Scrum projecten.

### 3.3.3 Implementaties

#### 3.3.5.1 Kanban

Kanban is als eerste methodologie geïmplementeerd in het bedrijf als deel van de stageopdracht. Het werd na een korte onderzoeksfase geïntroduceerd bij de start van de stageopdracht met als hoofddoel om de lopende projecten in kaart te brengen. Om te beginnen aan projectmanagement was er eerst een lijst nodig van de lopende projecten maar deze was niet aanwezig. Het Kanban bord heeft als oplossing gediend om die lijst te verkrijgen. Om deze te maken en een snel overzicht te verkrijgen op de huidige taken werd een Kanban bord op de muur geplaatst.

Figuur 10 stelt voor hoe het whiteboard van Firewolf Studios eruit zag tijdens de stageperiode. Er zijn 3 verticale kolommen van kaartjes aanwezig met de namen: to do, in progress en done. Taken die nog uitgevoerd moeten worden komen in de eerste rij (to do), taken die reeds worden uitgevoerd in de tweede rij (in progress) en taken die afgewerkt zijn in de laatste rij (done). Ook is het bord horizontaal verdeeld in rijen die de verschillende projecten moeten voorstellen. Het bord kent geen vierde kolom voor taken die uitgevoerd zijn maar nog getest moeten worden. De teamleden testen hun eigen uitgevoerde taken zelf en er is geen aangewezen persoon die testen uitvoert.

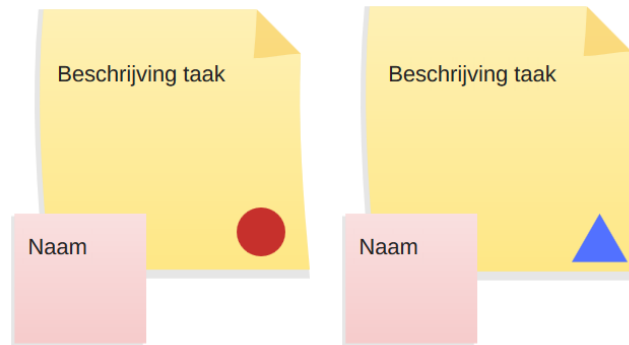


Figuur 10: Kanban toegepast op Firewolf Studios in eerste fase

De kaartjes hebben allemaal een legende door middel van een symbool in een bepaalde kleur: een rode cirkel staat bijvoorbeeld voor een technische taak, een blauwe driehoek voor een taak in verband met 3D. Indien er voor een game bijvoorbeeld een 3D-model moet worden gemaakt van een computer, krijgt de taak een blauwe driehoek als symbool. Zo is het voor het team makkelijk om te zien welke kaartjes aan welke teamleden gericht zijn.

In een volgende fase wordt er meer specifiek aangeduid welke taak uitgevoerd wordt door wie. Kleine kaartjes worden toegevoegd aan de grote kaarten met daarop steeds de namen van de personen die een taak gaan uitvoeren of al aan het uitvoeren zijn. Dit wordt voorgesteld in figuur 11.





Figuur 11: Kanban voorstelling post-it toegepast binnen Firewolf Studio

De volgende fase binnen Kanban is het limiteren van *work-in-progress* ofwel WIP. Dit staat voor de hoeveelheid werk waar men momenteel actief mee bezig is. Dit past niet goed binnen de studio: het is een kleine studio die genoodzaakt is om veel projecten tegelijk aan te nemen en uit te werken. Door een groot aantal projecten tegelijk te moeten uitvoeren in een klein team, bleek het niet handig te zijn om WIP te gaan limiteren.

De eerste implementatie van het Kanban bord was een fysiek bord op de muur. Er werd hiervoor gekozen omdat dit het makkelijkst zichtbaar was. Er moest niet naar een website genavigeerd worden om te zien welke projecten er allemaal lopen, een blik op de muur volstond. Deze manier van gebruik van Kanban is ook toegankelijker naar niet technische mensen binnen een team toe, zoals bijvoorbeeld artiesten die vaker voorkomen als teamleden binnen gamedevelopment. Tijdens de implementatie van een aantal issue-tracking- en projectmanagement tools binnen het onderzoek (verder beschreven in het tweede gedeelte van dit eindwerk), is dit overgegaan naar het gebruik van een digitale versie. Veel van deze tools boden het standaard aan als deel van hun software.

### 3.3.5.2 Scrum

Binnen Firewolf Studios wordt er reeds iteratief gewerkt voor sommige projecten. Er zijn een aantal projecten waarbij de feedback van de klant cruciaal is. Om te voorkomen dat er iets ontwikkeld wordt wat de klant niet wilt, wordt er daarom op regelmatige tijdstippen feedback van de klant verwacht. Dit dwingt het bedrijf tot het maken van een game in iteraties: het product is van het begin bruikbaar opdat de klant het kan testen en het krijgt per iteratie nieuwe *features*.

Het werken in iteraties is iets wat het bedrijf kan gaan toepassen binnen al hun projecten. Ze verkiezen er wel niet voor om Scrum tot in detail en formeel te gebruiken. Binnen Scrum wordt er bijvoorbeeld dagelijks gedaan aan een stand-up meeting waar er besproken wordt wat er die dag gedaan moet worden. Aangezien iedereen binnen het team samen op één kantoor zit en er makkelijk gepraat wordt lijkt dit voor het team overbodig. Het zou ook moeilijk zijn omdat veel medewerkers halftijds werken. Maar net doordat sommigen halftijds werken zou het nuttig kunnen zijn geweest indien het bedrijf ervoor zou kiezen om bijvoorbeeld 's morgens en 's middags een korte stand-up te houden.

Ook wordt er binnen Scrum een scrum master en een product owner aangeduid. Aangezien het team maar enkele medewerkers telt, is dit overbodig.

### 3.3.4 Resultaten

Eén van de doelstellingen binnen mijn stageopdracht was het verbeteren van projectmanagement binnen het bedrijf. Dit is gerealiseerd door het bedrijf te laten kennis maken met agile principes zoals Kanban en Scrum. Deze kennis wordt verder ondersteund door het gebruik van een projectmanagement- en issuetrackingtool, wat verder wordt aangehaald in deel 2 van dit document. Door deze tool zijn ze afgestapt van het gebruik van fysieke checklists en hebben ze het vervangen door een goede digitale geheugensteun voor alle projecten die lopen in het bedrijf en de bijhorende gegevens. Indien het bedrijf in de toekomst uitbreidt of nieuwe stagiairs aanneemt, dan kan deze tool hen op weg helpen binnen het bedrijf. Ook hebben ze door deze tool nu de mogelijkheid om aan timetracking te doen. Op die manier kunnen ze naar projecten in de toekomst toe beter inschatten hoeveel tijd bepaalde taken gaan innemen.

## 3.4 Bijkomende taken

Buiten het zoeken naar mogelijke verbeteringen binnen de testprocedure en het projectmanagement binnen het bedrijf, zijn er ook andere taken uitgevoerd binnen de stage die passen binnen het profiel van een softwaremanager. Zo is er bijvoorbeeld meegewerkt aan het opstarten van een nieuwe projecten en had de stagiaire inspraak op de ontwikkeling van veel lopende projecten.

### 3.4.1 Meewerken aan een nieuw project: Firewolf Game Builder

Een groot deel van de stageperiode is er tijd gespendeerd aan het meehelpen opstarten van een nieuw project binnen de studio. In de eerste weken ging dit vooral over brainstormen, ideeën rond het project bespreken en concreet maken. Hierna moesten deze ideeën in kaart gebracht worden in een document als voorbereiding op het ontwikkelen van een eerste prototype van het product. Uit deze nood is een SRS document ontstaan, wat ongeveer twee weken van de stageperiode in beslag heeft genomen. Uit het meehelpen opstarten van dit project heeft de stagiaire veel ervaring kunnen opdoen.

#### 3.4.1.1 Brainstormen

Binnen het bedrijf is er het idee om intern een eigen, nieuw project op te starten. In dit nieuwe project speelt het bedrijf met het idee om niet langer steeds een game te ontwikkelen voor een specifieke klant, maar een algemener project te maken. Dit algemener project moet een platform worden waarop de eindgebruiker zelf zijn eigen games kan gaan maken. Het is nog steeds gericht naar bedrijven toe en de focus ligt nog steeds op serious games. Alleen zullen de bedrijven nu zelf hun spel kunnen gaan ontwikkelen naar hun specifieke eisen.

Zo een platform moet bijgevolg wel enorm gebruiksvriendelijk zijn. Er wordt niet verwacht dat de eindgebruiker die games gaat ontwikkelen binnen dit platform een technische achtergrond heeft.

Bij de aanvang van de stageperiode was dit project nog maar net in ontwikkeling en heeft de stagiaire de mogelijkheid gekregen uitgebreid mee te denken in brainstormsessies en heeft de stagiaire eigen inbreng en ideeën over dit soort project kunnen leveren. Dit was vooral nuttig omdat er op dat moment enkel een idee was en er nog veel werd nagedacht over specifieke requirements er rond.

### 3.4.1.2 Pitch

Het bedrijf koos ervoor om in een eerste fase van dit project onderzoek te doen naar de mogelijkheden die het heeft. Dit door middel van studenten aan de Hogeschool PXL een prototype van hun idee te laten ontwikkelen als opdracht voor hun researchproject.

Om de studenten die konden kiezen hieraan deel te nemen te informeren, heeft de stagiaire een pitch ontwikkeld en heeft deze het idee zelf mogen voorstellen voor de studenten.

### 3.4.1.3 SRS

Belangrijk voor een team dat de ontwikkeling van zo een project start, is dat er heel duidelijke specificaties zijn en dat het hele team het project op dezelfde manier heeft begrepen. Het is heel belangrijk dat ieder lid van het team het doel van het project door en door begrijpt en genoeg inzicht heeft in het geheel.

Om de studenten die zullen werken aan het prototype van dit project op weg te helpen, is er een SRS document opgesteld. Dit document bevat een grondige uitleg over hoe het product eruit moet zien. Het heeft een minder technisch gedeelte dat de uitleg over het project in tekst bevat samen met voorbeelden van hoe de *interfaces* binnen het project er voor de eindgebruiker zullen uitzien. Een ander gedeelte bevat de meer technische specificaties van het project, voorgesteld door diagrammen en *use cases*.

Door de ontwikkeling van dit document kwamen enorm veel vragen naar boven. Zoals: “Hoe zal een game gemaakt binnen de tool tot bij alle eindgebruikers komen?” en “Welke *features* van het product horen op de applicatie thuis en welke eerder op een bijhorende website?”. Dit document heeft ervoor gezorgd dat het bedrijf over bepaalde onderwerpen dieper moest gaan nadenken en beslissingen moest gaan maken. Het in kaart brengen van de ideeën en vereisten van het bedrijf heeft geleid tot een heel specifiek concept voor een nieuw product.

Zo is er binnen het document heel specifiek uitgewerkt welke soorten gebruikers er zouden kunnen bestaan in het systeem, zijn er specifieke functies uitgelegd (zoals het beheren van assets, het aanmaken van games, het kopen van licenties) en zijn de onderzoeks-topics waar de studenten zich op kunnen gaan focussen, duidelijk geworden.

Bijlage B bevat een selectie van enkele pagina's uit dit document.

### 3.4.1.4 Begeleiden IT-project

Na het helpen opstarten van het project is de stagiaire het project ook blijven begeleiden. Tijdens de ontwikkeling van het prototype heeft de stagiaire de groep begeleid. Aangezien de stagiaire het product en de eisen van het bedrijf ondertussen door en door kende, stond de stagiaire in voor het beantwoorden van de vragen van het team. De studenten konden altijd bij de stagiaire terecht via Slack voor vragen of voor feedback op hun documenten of opgeleverd werk voor hun project.

Ook werd er minstens twee keer per week een bezoek gepland bij het team. Meestal gebeurde dit aan het begin van de week om de sprint planning te bespreken, en opnieuw op het einde van de week om de voortgang van het team te monitoren en bij te sturen. Om deze informatie ook te kunnen communiceren met de medewerkers van Firewolf Studios zijn er projectstatusrapporten opgesteld. Hierin werd beschreven wat het team gerealiseerd had in de afgelopen week. Ook werden er vragen van het team die de stagiaire moest doorgeven aan de medewerkers van Firewolf Studios in beschreven.

## 3.4.2 Overige brainstormsessies

### 3.4.2.1 Bezoek SDWorx

De stagiaire heeft ook de kans gekregen mee te werken aan een brainstormsessie op locatie. SDWorx, gelegen te Antwerpen was op zoek naar een oplossing om de vacature pagina van hun website interactiever te maken. Ook waren ze op zoek naar een product om nieuwe medewerkers tijdens welkomstdagen te helpen kennis maken met het bedrijf. Hiervoor had men het idee om een soort game te ontwikkelen met een quiz over het bedrijf.

## 3.5 Conclusie stageopdracht

Uit de resultaten van beide delen van de stageopdracht kan er geconcludeerd worden dat de doelstellingen bereikt zijn. Het bedrijf heeft kennis gemaakt met nieuwe testmethodes binnen gamedevelopment. Het bedrijf heeft ondervonden hoe ze kunnen testen in Unity zelf, hoe ze kunnen testen met Unium en Jasmine en hoe ze scripts kunnen schrijven in Python met PyAutoGUI om muis en toetsenbord invoer te kunnen automatiseren. Unium en PyAutoGUI zijn manieren van geautomatiseerd testen. Het bedrijf kan de toepassing hiervan gebruiken om de kwaliteit van hun games te ondersteunen naar de toekomst toe. Ook hebben ze kennis gemaakt met enkele agile methodologieën (Kanban en Scrum), wat hun visie en werking binnen toekomstige projecten kan ondersteunen.

Deze kennis zal hun zeker ondersteunen in het gebruik van hun gekozen projectmanagement- en issuetrackingtool. Dit wordt uitgebreid beschreven in het volgende deel van het eindwerk. Door het gebruik van deze tool zijn ze afgestapt van het gebruik van fysieke checklists en hebben ze het vervangen door een goede digitale geheugensteun voor alle projecten die lopen in het bedrijf en de bijhorende gegevens. Indien het bedrijf in de toekomst uitbreid of nieuwe stagiairs aanneemt, dan kan deze tool hen op weg helpen binnen het bedrijf. Ook hebben ze door deze tool nu de mogelijkheid om aan timetracking te doen. Op die manier kunnen ze naar projecten in de toekomst toe beter inschatten hoeveel tijd bepaalde taken gaan innemen.

## 4. Reflectie stageopdracht

Tijdens mijn stageperiode bij Firewolf Studios heb ik veel bijgeleerd. Uit persoonlijke interesse voor gamedevelopment heb ik ervoor gekozen mijn stageopdracht bij hun uit te voeren. De stageopdracht heb ik dan ook zelf aangehaald. Het heeft mij de kans gegeven om iets bij te leren uit mijn eigen interessegebied en tegelijkertijd mijn opgedane kennis uit mijn opleiding toe te passen. Ik zag het toepassen van de kennis uit mijn opleiding in deze sector daarom als een uitdaging naar mezelf.

Tijdens mijn stageopdracht heb ik bij het testen van de games veel vrijheid gekregen om onderzoek te doen wat ik heel erg apprecieer. Ik heb uitgebreid de kans gekregen om op zoek te gaan naar verschillende manieren om te testen, te onderzoeken welke testmethodes er voorkomen binnen gamedevelopment, te onderzoeken op welke manieren ik testen kon gaan automatiseren en te onderzoeken wat Unity (de *gamedevelopmentengine* naar keuze van het bedrijf) zelf aanbiedt om games te testen. Tijdens de lessen op de hogeschool hebben we enorm veel theorie gezien in verband met testen maar we hebben deze nooit praktisch toegepast in de zin dat we nooit testen geschreven hebben in de vorm van code. We hebben er wel geleerd hoe we FTT's en testcases opstellen en ontwerpen. Tijdens mijn stageperiode heb ik dan wel uitgebreid de kans gekregen mezelf hierin te verdiepen waar ik veel uit heb bijgeleerd.

Projectmanagement is altijd iets geweest waar ik de meeste voldoening uit haalde tijdens mijn opleiding. De rol van projectleider tijdens het IT-project in het derde jaar van onze opleiding, nam ik heel graag op mij en ik voerde deze rol graag uit. In de eerste weken van mijn stageperiode heb ik dan ook Firewolf Studios geholpen het projectmanagement binnen het bedrijf te optimaliseren. Hieruit ben ik dan onderzoek gaan doen over agile methodologieën binnen gamedevelopment, en daarop volgend ben ik onderzoek gaan doen naar projectmanagement- en issuetrackingtools die het bedrijf hierbij kunnen ondersteunen. Uit mijn interesse voor projectmanagement is ook het begeleiden van een IT-project tijdens de stageperiode gegroeid. Firewolf Studios had namelijk tijdens mijn stageperiode een eigen IT-project voor de Hogeschool PXL. Het was een leuke en leerrijke ervaring om nog eens zo een project mee te maken, maar dit keer vanuit het perspectief van de opdrachtgever. Bij de start van het IT-project heb ik het idee ervan mogen pitchen voor alle studenten van het derde jaar die dit vak opnamen. Ik had nog niet eerder de kans gekregen om te pitchen voor zoveel mensen. Ik ben blij deze ervaring gehad te hebben.

Een andere bijkomende opdracht hierbij was het ontwikkelen van een SRS (software requirements specification). Dit was een document dat ik geschreven heb waarin alle informatie die de studenten die aan het IT-project moesten gaan werken, bevatte. Het bevatte algemene informatie, *mock-ups*, *use cases* en een lijst van *features* die het product moest hebben. Er is veel tijd ingekropen. Een deel van deze tijd ontstond doordat er steeds bij het nalezen van het document door het bedrijf vragen ontstonden. Deze vragen leidden vaak tot een aanpassing binnen het idee of concept waardoor een groot deel van het document moest worden aangepast. Ik had deze tijd meer kunnen gebruiken tijdens het onderzoek naar verschillende manieren om games te testen, maar ik heb hier wel een les uit getrokken in verband met communicatie.

## II. Onderzoekstopic

### 1. Onderzoeksvraag

#### 1.1 Omschrijving

Kunnen er issuetracking- en projectmanagementsystemen gebruikt worden in een gamedevelopment studio? Zo ja, welke zouden hier dan voor gebruikt kunnen worden?

Binnen het onderzoek wordt er gezocht naar een passend issuetracking en projectmanagement systeem voor Firewolf Studios. Er wordt eerst aan de hand van een literatuuranalyse bekeken op welke manier projectmanagement voorkomt binnen gamedevelopment. Hierna worden een aantal tools geselecteerd op basis van een set van requirements uit het bedrijf. Deze tools worden geïmplementeerd en met elkaar vergeleken. Het resultaat van het onderzoek is dus een tool die past binnen Firewolf Studios, ofwel een kleine gamedevelopmentomgeving.

#### 1.2 Link met stageopdracht

Tijdens de stageopdracht wordt er aan softwaretesten gedaan en aan projectmanagement. Het projectmanagement gedeelte wordt door middel van Kanban uitgewerkt op een fysiek whiteboard op de muur, maar naar de toekomst toe zou dit beter online gebeuren aangezien de meeste medewerkers halftijds werken of geoutsourcet zijn. Ze zouden door middel van een projectmanagement tool hun taken ook online kunnen raadplegen.

Tijdens het uitvoeren van testen komen er issues naar boven. Voor deze issues moet er een plaats zijn waar deze opgeslagen kunnen worden. Een issuetrackingsysteem kan hier voor een oplossing zorgen. Het bijhouden van issues maakt het makkelijker een overzicht te bewaren, en op deze manier wordt er ook kennis bijgehouden over hoe een issues die zich al eens voor heeft gedaan al opgelost werd.

### 2. Onderzoeksmethode

#### 2.1 Aanpak

Om aan het onderzoek te beginnen zal er eerst een algemeen onderzoek nodig zijn naar welke tools er allemaal beschikbaar zijn op de markt. Om een goede vergelijking te maken tussen verschillende tools, wordt er eerst een lijst opgesteld van ongeveer 50 tools om een ruime verscheidenheid aan tools te hebben.

Om vervolgens verder te onderzoeken welke tools hieruit de beste zijn, wordt er eerst aan requirements gathering gedaan. Om een tool te kiezen die zo juist mogelijk past binnen het plaatje van dit bedrijf, moeten hun noden eerst in kaart worden gebracht. Zo wordt er een set van requirements specifiek voor dit bedrijf opgesteld. Dit wordt gedaan door middel van interviews met de uiteindelijke eindgebruikers.

## 2.2 Analyse

Aan de hand van de requirements van het bedrijf kan ik de tools verder gaan analyseren. Zo kan ik iedere tool een bepaalde score gaan geven op basis van deze requirements. Afhankelijk van hun uiteindelijke score op deze requirements komen de tools uiteindelijk op een bepaalde plaats in de lijst terecht. Op deze manier worden er dus al een aantal tools uit de lijst gefilterd. De tools met de hoogste score worden in gebruik genomen tijdens de stageperiode. Op deze manier worden de tools vergeleken. Dit wordt nader besproken in hoofdstuk 4.1.2 en is beschikbaar in bijlage C.

Aan de hand van een planning gemaakt na de analyse van de tools zullen er enkele tools vergeleken worden telkens gedurende een aantal weken. Tijdens de implementatie worden de gebruikers bijgestaan om de tool zo duidelijk mogelijk te leren kennen.

Na het in gebruik nemen van de tool voor een aantal weken zal er op het einde steeds een evaluatie volgen van de tool. Dit aan de hand van een vragenlijst: zo krijgt de tool een score van de eindgebruikers. Deze vragenlijst wordt nader besproken in hoofdstuk 4.1.3 en is beschikbaar in bijlage D.

Na de evaluatie van de verschillende tools is er nog een kleine periode beschikbaar waarin de uiteindelijk gekozen tool geïmplementeerd wordt en er een laatste evaluatie gedaan zal worden. De tools worden hierin ook onderling met elkaar vergeleken.

## 3. Literatuurstudie

### 3.1 Aanleiding tot literatuurstudie

Vooraleer aan het onderzoek van deze tools te beginnen, wordt er een breder onderzoek uitgevoerd. Het gebruik van de tools die in het onderzoek aangehaald worden, vloeit meestal door vanuit noden binnen projectmanagement of algemeen management van een project. Veel van deze tools bieden op zichzelf dan nog hulp binnen projectmanagement door functionaliteiten aan te bieden op dit vlak. Dit gaat vaak over agile principes en kan gaan over het gebruik van een Kanban bord, of het werken met iteraties zoals bij scrum wordt gezien, binnen de tool.

Daarom wordt er eerst een literatuurstudie uitgevoerd waarin wordt nagegaan hoe het zit met het managen van gamedevelopmentprojecten, en meer specifiek hoe het zit met het gebruik van agile methodologieën binnen gamedevelopment, aangezien veel van de tools in het onderzoek hiervoor mogelijkheden bieden.

### 3.2 Paper 1: *Game development software engineering process lifecycle: a systematic review* door S. Aleem, L. F. Capretz en F. Ahmed

#### 3.2.1 Overzicht

De paper *Game development software engineering process lifecycle: a systematic review* is een soort van verzameling van vorige onderzoeken binnen het onderwerp van gamedevelopmentprocessen, opgesteld door S. Aleem, L. F. Capretz en F. Ahmed. Hij geeft een algemene uitleg over de vergelijking van gamedevelopment en traditionele softwaredevelopment en tast aan de hand van een literatuurstudie de maturiteit van dit domein af.

Door middel van dit onderzoek worden er nieuwe onderzoeksvragen opgewekt en dit is ook het uiteindelijke doel van de paper. In de paper beseffen de onderzoekers dat er nog veel gebieden binnen gamedevelopment niet onderzocht zijn en wordt geprobeerd het enthousiasme op te wekken bij nieuwe onderzoekers om dit gebied toch te verkennen.

Enkel de eerste gedeelten van de paper worden hieronder besproken aangezien zij het meest relevant zijn naar de onderzoeksvraag toe. De rest van de paper focust zich op het verzamelen van verdere onderzoeksvragen binnen gamedevelopment in het algemeen.

### 3.2.2 Bespreking

Gamedevelopment is een technologie die steeds populairder wordt: steeds meer mensen spelen games. Het wordt ook steeds belangrijker: hierbij valt de opkomst van 'serious games' op. Gamedevelopment is iets heel complex aangezien het heel multidisciplinair is. Binnen de development van een game komen zeer uiteenlopende disciplines aan bod, van tekenkunst tot 3D modellering, animatie en zeker ook programmeren. De multidisciplinaire aard van gamedevelopment is het voornaamste verschil tussen gamedevelopment en softwaredevelopment. Maar ook al is er een groot verschil, toch is het ook een softwaredevelopmentproces aangezien het uiteindelijk afgeleverde product ook software is.

Voor gamedevelopment is er niet een vast 'proces' of verloop, maar de onderzoekers die al getracht hebben om een proces voor gamedevelopment in kaart te brengen komen met verscheidene opties naar voren. Er worden 5 resultaten van verschillende onderzoekers aangekaart, die elk het gamedevelopmentproces voorstellen met verschillende fases. De paper hanteert zelf maar 3 fases binnen het proces, in tegenstelling tot de vermelde onderzoekers die veel meer fases aankaarten.

Binnen deze paper kent het gamedevelopmentproces een preproductiefase waar de haalbaarheid van een spel onderzocht wordt en waar er aan requirements engineering gedaan wordt als eerste fase. Als tweede fase een productiefase, waarin de planning van het project gebeurt, de documentatie uitgeschreven wordt en de eigenlijke game in development gaat. En als laatste een derde fase, de postproductiefase waarin getest wordt, het spel op de markt wordt gezet en er geadverteerd wordt.

Men is op zoek naar een algemene 'development-methode', een systematische procedure om een doel te bereiken. In dit geval: een werkbaar product te creëren binnen tijd en budget. Binnen traditionele softwareontwikkeling zijn er een aantal methodologieën om dit doel te bereiken zoals *waterfall*, maar ook agile methodologieën zoals Scrum en Kanban. De paper voert onderzoek uit naar de implementatie van deze technologieën binnen gamedevelopment.

Bij *waterfall* worden er na de preproductiefases activiteiten uitgevoerd volgens het *waterfall* principe. Dit betekent dat de activiteiten eerst verdeeld en gegroepeerd worden en vervolgens aan verscheidene teams overgedragen worden. Binnen deze methode is het nog moeilijk om enige activiteit terug te draaien en ongedaan te maken.

Agile is gebaseerd op iteraties. Het productieproces wordt opgedeeld in kleine iteraties en de focus ligt bij de meest cruciale *features*. Het team zit steeds volledig samen bij iedere iteratie en na elke iteratie hoort er contact met de klant te zijn. De meest voorkomende agile methodes die blijken voor te komen binnen gamedevelopment zijn XP (extreme programming), rapid prototyping en Scrum.



Een foute maar vaak voorkomende assumptie was dat het gamedevelopmentproces een *waterfall* methode moest volgen, maar recent komt men steeds meer tot de conclusie dat het incrementeel zou moeten zijn.

Het gamedevelopmentproces wordt veelvuldig in vraag gesteld en men probeert op veel manieren een proces en de types van practices die gebruikt worden in kaart te brengen.

Dit besluit de introductie en achtergrond van de paper. Verdere uitleg in de paper gaat over de gedeeltes binnen gamedevelopment waar nog onderzoek nodig is.. [9]

### 3.2.3 Besluit

Het hoofddoel dat deze paper is eigenlijk het aankaarten van de nood van research binnen het topic, er wordt meermaals aangekaart dat er binnen gamedevelopment veel vragen zijn naar mogelijke methodologieën. Er wordt slecht kort aangehaald dat men vaak denkt dat het de bedoeling is dat men *waterfall* toepast als methodologie, maar dat men binnen gamedevelopment steeds meer iteratief begint te werken.

## 3.3 Paper 2: How are agile methods and practices deployed in video game development? A survey into Finnish game studios door J. Koutonen en M. Leppänen

### 3.3.1 Overzicht

De paper *How are agile methods and practices deployed in video game development? A survey into Finnish game studios* door J. Koutonen en M. Leppänen, gaat na hoe het zit met het gebruik van verscheidene agile methoden binnen enkele Finse gamedevelopmentstudio's. onderzoek is gebeurd aan de hand van een zorgvuldig opgestelde enquête die niet alleen vragen bevatte in verband met agile methoden, maar ook algemene vragen gericht aan de bedrijven. Er werd hun bijvoorbeeld ook gevraagd hoe oud en hoe groot hun bedrijf was om de latere analyse van de resultaten bij te staan.

### 3.3.2 Bespreking

De game-industrie is heel snel heel groot aan het worden, maar de industrie zit met veel uitdagingen. Zo verwacht de eindgebruiker bijvoorbeeld steeds meer van de games die hij of zij speelt.

Gamedevelopment is een veeleisende sector aangezien projecten die uitgevoerd worden steeds bestaan uit multidisciplinaire teams. Ook is het een risicovolle en competitieve sector, uitgevers van games die investeren in de development ervan weten niet of het spel waar ze in gaan investeren wel succesvol zal zijn. De markt verandert ook voortdurend.

Aangezien gamedevelopment zo complex is zijn er vaak problemen in bijvoorbeeld planning en coördinatie binnen projecten. Binnen traditionele software ontwikkeling wordt hier hulp geboden door middel van het toepassen van verscheidene methodes. En aangezien gamedevelopment ook een proces is waar software ontwikkeld wordt, wordt de effectiviteit en het gebruik van deze methodologieën binnen gamedevelopment in vraag gesteld.

Commerciële games worden vaak ontwikkeld aan de hand van *milestones*, deze *milestones* zijn van belang als het gaat over contracten met de uitgevers. Traditioneel is gamedevelopment gebaseerd op *waterfall* en varianten hiervan, maar nu kiezen veel teams een iteratief proces met prototypes zodat ze kunnen testen in vroegere fases van het project en snel veranderingen binnen het project kunnen doorvoeren. Dit komt terug in agile methodes zoals scrum, XP, Lean en Kanban. Het begrijpen van hoe agile methoden toegepast worden op gamedevelopment en de gevolgen en impact ervan is een weinig onderzocht domein dat volgens dit artikel extra aandacht verdient.

In het onderzoek blijkt allereerst dat veel van de bedrijven die meededen jonge en kleine bedrijven waren, waar PC en mobiele games het meest populair waren en waar projecten vaak slechts een jaar in development waren. De meeste bedrijven hadden maximaal 3 projecten die tegelijk liepen.

Uit resultaat van het onderzoek blijkt dat de meeste bedrijven die agile methoden toepasten, scrum toepasten als methodologie. Gevolgd door Lean en daarna Kanban. Het blijkt dat hoe jonger de studio is, hoe minder agile ze werkt wat verrassend is aangezien agile steeds meer opkomend zou zijn. De uitleg zou te vinden kunnen zijn in de voorkeur naar ad-hoc werken voor jonge bedrijven.

Als het ging over specifieke beoefeningen (agile practices in plaats van methoden), dan kwamen scrum practices het meeste naar boven. Het dagelijks scrummen en werken met sprints bleek hier het meest populair te zijn.

Enkele practices uit XP werden ook redelijk vaak toegepast, het ging dan over het werken met een cross functioneel team (wat niet vreemd is als er gewerkt wordt met een multidisciplinair team) en continue integratie. Kanban kwam weer slechts weinig voor. Het blijkt dat helemaal geen studio bij hield hoe lang een taak in behandeling was. Met andere woorden, hoe lang een taak als WIP (work in progress) werd gezien.

Verdere resultaten van het onderzoek gingen over de impact van agile binnen deze studio's. De studio's kaartten vooral verbetering aan in communicatie, kwaliteit van het product en verbeteringen binnen het implementeren van nieuwe *features*.

Als conclusie blijkt dus dat op 1 studio na, alle studio's wel op de een of andere manier in aanraking kwamen met agile methodes. Dit hoefde niet altijd een traditionele methode te zijn, vaak was het een samenvoeging van verschillende practices afhankelijk van de noden binnen het bedrijf. [10]

### 3.3.3 Besluit

De paper bevestigt dat de opkomst van agile binnen traditionele softwaredevelopment ook aanwezig is in gamedevelopment. Het besluit dat Scrum daarin het meest voorkomend is en Kanban het minst, maar dat veel bedrijven de mogelijk methodologieën en practices op hun eigen manier toepassen afhankelijk van de noden van het project of bedrijf. Ze houden het dus niet op 1 methodologie voor het hele bedrijf maar experimenteren met mogelijkheden.

## 3.4 Paper 3: *Is agile out there? Agile practices in gamedevelopment* door F. Petrillo en M. Pimenta

### 3.4.1 Overzicht

De paper *Is agile out there? Agile practices in gamedevelopment* door F. Petrillo en M. Piment gaat aan de hand van post mortem analyses na of agile voorkomt binnen gamedevelopment of niet. Met post-mortem analyses bedoelt men analyses die gebeuren na de ontwikkeling van een product of na de afloop van een project. Veel gamedevelopment teams (in tegenstelling tot teams binnen de traditionele software development) maken na afloop van een project een analyse op waarin beschreven staat wat er goed ging en wat er fout ging in het project, zodat het bedrijf (en ook anderen) hier uit kunnen leren en er mogelijkheid is tot verbetering. Aan de hand van analyses van dit soort documenten, wordt onderzocht of er binnen gamedevelopmentprojecten agile methoden toegepast zijn geweest.

### 3.4.2 Bespreking

Gamedevelopment is complex en multidisciplinair, maar ondanks de vele uitdagingen is het toch de grote leider binnen de entertainment industrie. Om dit resultaat te bereiken lijkt het logisch dat er goede practices toegepast geweest moeten zijn.

Binnen het onderzoek gaat men niet zo zeer op zoek naar het gebruik van agile methodologieën, maar bekijkt zoekt men eerder naar het gebruik van kleinere practices die binnen agile ook voorkomen.

*Best practices* die binnen traditionele softwaredevelopment voorkomen blijken inderdaad ook voor te komen in gamedevelopment. De practices kunnen verbonden worden aan methodes zoals Scum en XP. Teams gebruiken steeds meer agile practices uit eigen instinct. Ze gebruiken het informeel en misschien zonder dat ze er zich bewust van zijn. [7]

### 3.4.3 Besluit

De paper toont aan dat er wel degelijk gebruik gemaakt wordt van agile methodologieën binnen gamedevelopment, maar dat het informeel gebeurt. Het kaart aan dat aangezien veel problemen die in traditionele software ontwikkeling voorkomen ook in gamedevelopment voorkomen, het niet meer dan normaal is dat dezelfde oplossingen gebruikt gaan worden.

## 3.5 Vergelijking en conclusie

De papers tonen allemaal aan dat agile opkomend is op verscheidene manieren binnen gamedevelopment. Het is niet altijd even tastbaar, sommige bedrijven zijn zich er van bewust dat ze agile methodes toepassen terwijl anderen er gewoon instinctmatig gebruik van maken. Agile wordt toegepast afhankelijk van de noden van het bedrijf.

Aangezien het dus opkomend is, is het dus zeker waardevol om ook in dit onderzoek de mogelijkheden van agile werken aan te kaarten en verder te onderzoeken. Dit bijvoorbeeld door het gebruik van bepaalde *features* binnen de tools die onderzocht gaan worden, maar ook door het analyseren van voorbije en komende projecten. Voorbije projecten kunnen geanalyseerd worden op het gebruik van agile methodes, is er in het verleden reeds op een manier agile gewerkt? En op toekomstige projecten kan agile werken ook toegepast worden na een analyse van de noden van het bedrijf.

## 4. Resultaten

### 4.1 Selectie van tools

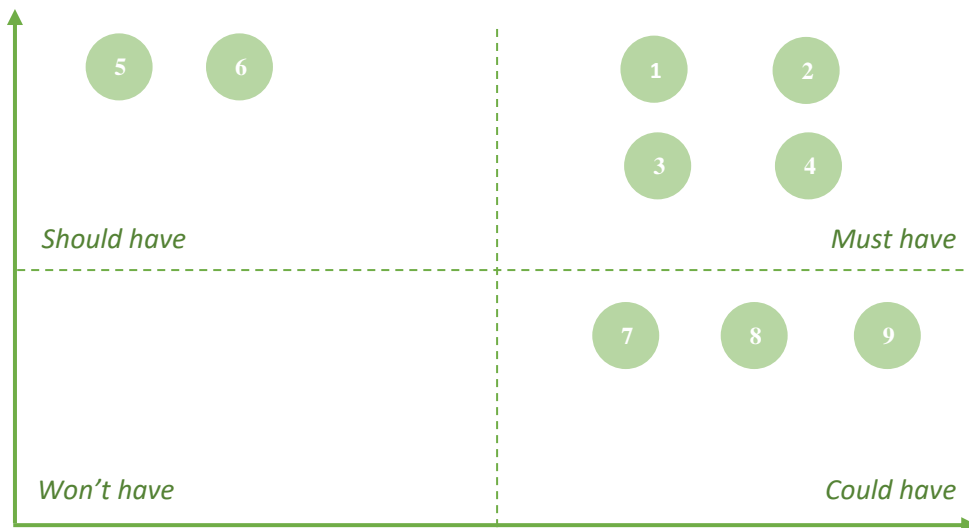
In de selectiefase van het onderzoek werd de markt onderzocht en werd er bekeken welke tools er eigenlijk allemaal beschikbaar zijn. Hierna werden deze tools beoordeeld om zo tot een keuze van tools te komen die geïmplementeerd kunnen worden. Deze tools worden dan effectief gebruikt en met elkaar vergeleken in een latere fase.

#### 4.1.1 Requirements

Vooraleer te beginnen aan het onderzoek, maakte het bedrijf zijn requirements bekend. Enkele requirements waren voor hun belangrijker dan andere. Hierdoor zijn deze requirements opgesteld in categorieën: *must have*, *should have* en *could have* net zoals bij een MoSCoW diagram. Bij een MoSCoW diagram is er nog een categorie *won't have*. Dit is niet van toepassing binnen dit onderzoek. Tabel 7 toont de requirements van het bedrijf en een bijpassend MoSCoW diagram in figuur 12 stelt de noden van het bedrijf ook nog eens visueel voor.

1. De tool heeft een mobiele app	MUST HAVE
2. De tool biedt een mogelijkheid tot tijdschrijven	
3. Data kan geïmporteerd worden vanuit een andere tool of geëxporteerd worden naar een andere tool	
4. De tool is beschikbaar op de cloud en moet niet zelf gehost worden	
5. Er kunnen meerdere projecten aangemaakt worden binnen de tool	SHOULD HAVE
6. Er is een duidelijk verschil tussen een 'bug' en een 'feature'	
7. De tool biedt mogelijkheden tot projectmanagement	COULD HAVE
8. De tool kan aan Git gekoppeld worden	
9. De tool kan met andere dingen zoals veel gebruikte IDE's gekoppeld worden	

Tabel 7: De vereisten van het bedrijf



Figuur 12: MoSCoW diagram van de vereisten

Vervolgens worden de vereisten van het bedrijf nog eens duidelijker uitgelegd:

1. Het bedrijf is op zoek naar een tool die ze op hun mobiele toestellen zoals smartphone en tablet ook kunnen raadplegen. Op vlak van besturingssysteem is er geen voorkeur, maar een tool die Android en iOS allebei ondersteund zou het hoogste scoren. Er zijn Android en iOS gebruikers binnen het team.
2. Een andere nood van het bedrijf is het kunnen bijhouden hoeveel tijd een medewerker aan eenzelfde taak werkt.
3. Het bedrijf denkt toekomstgericht en wil daarom zeker een tool kiezen waarvan de data geëxporteerd kan worden naar een andere tool indien er in de toekomst een betere tool op hun pad komt. Ideaal is een tool waarin er ook data van andere systemen kan geïmporteerd worden.
4. Men is op zoek naar een tool die beschikbaar is onder een SAAS licentie, hiermee bedoelt men dat men op zoek is naar een tool die gehuurd kan worden en op de cloud beschikbaar is. Men wil momenteel geen tijd spenderen aan het opzetten van een tool op een server, maar behoudt dit wel als een mogelijke piste naar de toekomst toe indien het zelf hosten voordelen met zich meebrengt.
5. Binnen het bedrijf lopen er vaak heel veel projecten tegelijk. Er zijn tools die een beperking hebben op het aantal projecten dat er tegelijkertijd kunnen bestaan in hun systeem, dit wil het bedrijf dus niet. Ideaal is een tool waarin er een ongelimiteerd aantal projecten aangemaakt kan worden.
6. Sommige tools gebruiken de begrippen 'feature' en 'bug' door elkaar of maken zelfs geen onderscheid. Het bedrijf vereist hiervoor een duidelijk onderscheid. Ideaal zou dus een tool zijn waar de velden aanpasbaar zijn naar de eigen noden.
7. Om projectmanagement naar de toekomst toe verder te zetten op een online platform zou het handig zijn dat dit reeds in deze tool geïmplementeerd is en er hiervoor geen tweede tool hoeft gebruikt te worden.
8. Het bedrijf zou het handig vinden een link te zien tussen issues in de tool en meldingen binnen *Git*. Het zou voor hun handig zijn op de tool snel te zien welke code aanpassingen vast hangen aan een issue of *feature* binnen *Git*.
9. Het zien van volgende taken en op te lossen issues binnen hun meest gebruikte IDE's wordt ook gezien als een interessante mogelijkheid.

### 4.1.2 Onderzoek naar mogelijke tools

Na het verzamelen van deze vereisten werden ze toegepast op een lijst van tools. Het verzamelen van deze lijst werd gedaan door rond te kijken op het internet. Veel tools kwamen terecht in de lijst door een reeds bestaande lijst op Wikipedia. [11] Maar ook een aantal tools zijn er in terecht gekomen door eigen kennis van het product of reclame.

De lijst bestaat uit 47 tools. Voor al deze tools is onderzocht of ze voldeden aan de vereisten van het bedrijf. Initieel werd dit bijgehouden door middel van een Excel sheet waar er in tekst vorm per vereiste van het bedrijf een beoordeling gegeven werd. Later werd dit duidelijker gemaakt aan de hand van kleuren. En om de tools uiteindelijk op een juiste manier te kunnen sorteren, is er gebruik gemaakt van een punten systeem. Per vereiste krijgen de tools een score op 10. Dit resulteert in een uiteindelijke score op 100. Aan de hand van deze uiteindelijke score is er een duidelijk top 5 van te testen tools. Deze lijst en de scores ervan zijn terug te vinden in bijlage C.

### 4.1.3 Evaluatieformulier

Om de tools te kunnen evalueren en een score te kunnen geven, wordt er een steekproef uitgevoerd door middel van een vragenlijst. Deze vragenlijst is terug te vinden in Bijlage B. De vragenlijst wordt ingevuld door beide ontwikkelaars binnen het bedrijf. Het evaluatieformulier wordt steeds ingevuld op de laatste dag dat de tool gebruikt wordt.

De antwoorden op deze vragen worden achteraf omgezet in numerieke scores. Zeer oneens zal een score van -2 opleveren, oneens een score van -1, neutraal zorgt voor een 0 score op de vraag. Indien de gebruiker het eens is, wordt de score +1. En voor een gebruiker die het zeer eens is wordt dat +2. Wanneer er geen antwoord wordt gegeven resulteert dat in een '/' in de tabel, deze heeft ook als waarde 0. Een tool die in totaal een score van 0 zou halen zou dus geen slechte tool zijn, maar een neutrale tool. Boven 0 bevat de tool positief en onder 0 is de tool minder geschikt.

Niet over elk requirement wordt er een vraag gesteld. Er worden enkel vragen gesteld over de requirements waarop de eindgebruiker een nuttige beoordeling kan maken. Op de vraag of de tool beschikbaar is op de cloud kan niet veel meer worden geantwoord dan ja of nee.

De tools krijgen een eindscore en een gemiddelde eindscore van beide participanten. Ook wordt er in de vergelijking op het einde van het onderzoek gekeken naar het percentage dat de tool scoort. Dit percentage wordt berekend aan de hand van de totaalscore en de maximumscore die een tool kan halen op de vragenlijst. Er zijn 36 vragen en de tool kan maximum 2 punten scoren op een vraag. Voor een gebruiker kan de totaalscore dus maximaal 72 zijn, en het eindtotaal maximaal 144.

De vragen zijn ingedeeld in categorieën. Categorie 1 en 2 stellen vragen over de interoperabiliteit met andere systemen. Categorie 1 stelt vragen over de verbinding met *Git* en 2 stelt vragen over het samenwerken tussen de tool en IDE's binnen het bedrijf. Categorie 3 gaat over security en stelt vragen over *access management* binnen de tool. Categorie 4, 5 en 6 gaan over functionaliteit en stellen vragen over het tijdschrijven, de duidelijkheid van het verschil tussen *bug* en *feature* en projectmanagement binnen de tool. Categorie 7 gaat over de portability van het systeem, meer specifiek: het stelt vragen over de beschikbaarheid van een mobiele app voor de tool. Categorie 8 stelt algemene vragen over de gebruiksvriendelijkheid en duidelijkheid van de tool.

#### 4.1.4 Keuze tools

De top 5 tools, als resultaat van de vergelijking uit hoofdstuk 4.1.2, gesorteerd op basis van hun score:

1. YouTrack
2. Perforce
3. FogBugz - Manuscript
4. SpiraTeam
5. Jira Software

Aangezien de stageperiode het niet toe liet alle tools grondig te testen, maakte het bedrijf een selectie uit deze top 5. Perforce viel bijvoorbeeld al snel uit de mand omdat de online versie ervan niet bestond in een gratis proefversie. Het bedrijf heeft de mogelijkheid niet om een versie te testen die moet werken op een eigen server. Deze tools werden nader bekeken, en daaruit volgde dan volgende keuze:

1. YouTrack
2. Jira
3. FogBugz - Manuscript

De eerste planning (tabel 8) bestond eruit de tools steeds twee weken te implementeren, afhankelijk van de beschikbare tijd tijdens het uitvoeren van de stageperiode. Hierna volgt de eerste opgestelde planning, en daarna de eigenlijke planning die ontstaan is door enkele keuzes door het bedrijf.

Naam tool	Begindatum	Einddatum	Aantal testdagen
<b>YouTrack</b>	Woensdag 25/10/2017	Dinsdag 14/11/2017	<b>15</b>
<b>Jira</b>	Woensdag 15/11/2017	Vrijdag 1/12/2017	<b>13</b>
<b>FogBugz - Manuscript</b>	Maandag 4/12/2017	Vrijdag 22/12/2017	<b>15</b>

Tabel 8: Planning implementatie tools

Volgens de originele planning wordt op deze manier elke tool minimum 13 dagen getest. Volgens deze planning blijft er nog een week over na de kerstvakantie om de uiteindelijke geïmplementeerde tool uitgebreid te configureren en implementeren. De eigenlijke uitvoering van het onderzoek ziet er anders uit (tabel 9):

Naam tool	Begindatum	Einddatum	Aantal testdagen
<b>YouTrack</b>	Woensdag 25/10/2017	Dinsdag 14/11/2017	<b>15</b>
<b>Jira</b>	Woensdag 15/11/2017	Vrijdag 1/12/2017	<b>13</b>
<b>FogBugz - Manuscript</b>	Maandag 4/12/2017	Dinsdag 5/12/2017	<b>2</b>
<b>SpiraTeam</b>	Woensdag 6/12/2017	Maandag 18/12/2017	<b>9</b>

Tabel 9: Uitvoering implementatie tools

FogBugz – Manuscript werd slechts twee dagen geïmplementeerd. Dit komt omdat er bij de implementatie van dit product heel snel bleek dat het toch niet was wat het bedrijf verwachtte, ookal scoorde het in het begin hoog op de requirements. Er is dan de keuze gemaakt om een andere tool binnen de lijst te kiezen. Er is dan voor SpiraTeam gekozen als vervanging.

Maar ook SpiraTeam kon volgens het bedrijf Jira niet evenaren, waardoor ook het gebruik van deze tool werd gestaakt. Op vraag van het bedrijf is het onderzoek na implementatie van deze tool afgerond, en kon de resterende tijd van de stageperiode gebruikt worden om Jira te configureren.

## 4.2 Resultaten implementatie

### 4.2.1 YouTrack

#### 4.2.1.1 Algemene informatie

YouTrack is een commerciële projectmanagement- en issuetrackingtool ontwikkeld door het bedrijf JetBrains. JetBrains is bekend door de ontwikkeling van hun vele IDE's zoals PhpStorm, IntelliJ en PyCharm. Ook zijn zij de bedenkers achter de programmeertaal Kotlin.

YouTrack kan gebruikt worden op de cloud, of het kan gehost worden op een eigen server.

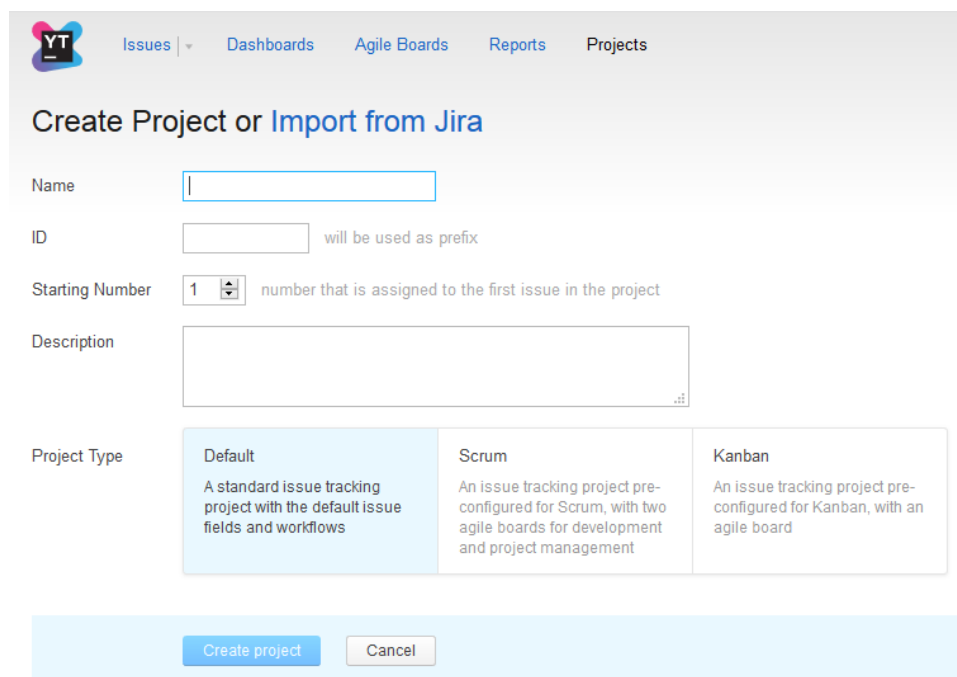
De focus van YouTrack ligt op query gebaseerde zoekopdrachten met *auto-completion*, groepen van issues tegelijk kunnen aanpassen, de attributen van een issue naar eigen wens kunnen aanpassen en aangepaste *workflows*. [12]

#### 4.2.1.2 Installatie en configuratie

Het opzetten van YouTrack op de Cloud is eenvoudig. Op de website van YouTrack kiest de gebruiker een naam voor zijn website en voert hij een e-mail adres in. Op dit adres krijgt de gebruiker vervolgens een mail met een link naar zijn website.

Wanneer deze gevolgd wordt ziet de gebruiker een pop-up waarin een account geregistreerd kan worden voor de website. Op de achtergrond wordt de website geladen.

Vervolgens krijgt de gebruiker al onmiddellijk de vraag of er een nieuw project gemaakt moet worden of dat de gebruiker iets wil importeren vanuit Jira.



Figuur 13: Een eerste project in YouTrack



Zoals te zien is in figuur 13 maakt YouTrack een onderscheid tussen Scrum en Kanban projecten of kan er gekozen worden voor een default project.

Wanneer er een issue aangemaakt wordt voor een van de drie opties, dan zijn de eigenschappen die overal ingesteld kunnen worden een prioriteit (priority) en een aangewezen persoon die de issue hoort op te lossen (assignee).

Wanneer er een Scrum project aangemaakt wordt, dan werkt YouTrack met sprints. Indien er een issue aangemaakt wordt binnen zo een project dan hebben we de keuze om een aantal scrum specifieke eigenschappen in te stellen. De huidige sprint kan aangeduid worden, er kunnen story points geschat worden voor de issue en een ideaal aantal dagen waarin de issue opgelost zou moeten zijn Er kan een type taak aangeduid worden. YouTrack biedt hier standaard bug, epic, user story en task aan. Alsook een status, hier hebben we de keuze open, in progress, to verify, done en duplicate.

Wanneer er een Kanban project aangemaakt wordt dan zijn er andere eigenschappen beschikbaar voor een issue. Figuur 14 toont een vergelijking hiervan. Binnen een Kanban project krijgt een issue een bepaalde stage. Een issue wordt bijvoorbeeld in de backlog gezet of komt terecht in een bepaalde kolom op het Kanban bord. Standaard biedt YouTrack hier als optie: backlog, develop, review, test, staging en done. Verder kan het een Kanban state hebbe, dit betekent dat het klaar is om opgenomen te worden door iemand of niet.

Een issue die aangemaakt wordt voor een default project kan een type hebben (bug, cosmetics, exception, feature, task, usability problem, performance problem, epic) een status (submitted, open, in progress, to be discussed, reopened, can't reproduce, duplicate, fixed, won't fix, incomplete, obsolete, verified), een subsysteem, een versie waarin deze issue verholpen is, de versies waarin deze issue voorkomt en in welke build de issue verholpen zal zijn.

Het valt op dat default meer standaard keuzes heeft voor de verschillende mogelijke eigenschappen tegenover Scrum en Kanban projecten binnen YouTrack.

Project	testscrum	Project	testkanban -	Project	testdefault -
Priority	Normal - <span>N</span>	Priority	Normal - <span>N</span>	Priority	Normal - <span>N</span>
Type	Task -	Assignee	Unassigned -	Type	Bug -
State	Open -	Kanban State	Ready to pull - <input type="checkbox"/>	State	Submitted -
Assignee	Unassigned -	Stage	Backlog -	Assignee	Unassigned -
Sprints	Unscheduled -			Subsystem	No Subsystem -
Ideal days	Not estimated			Fix versions	Unscheduled -
Story points	Not estimated			Affected versions	Unknown -
				Fixed in build	Next Build -

*Figuur 14: Vergelijking eigenschappen van issues in een scrum, kanban en default project in YouTrack*

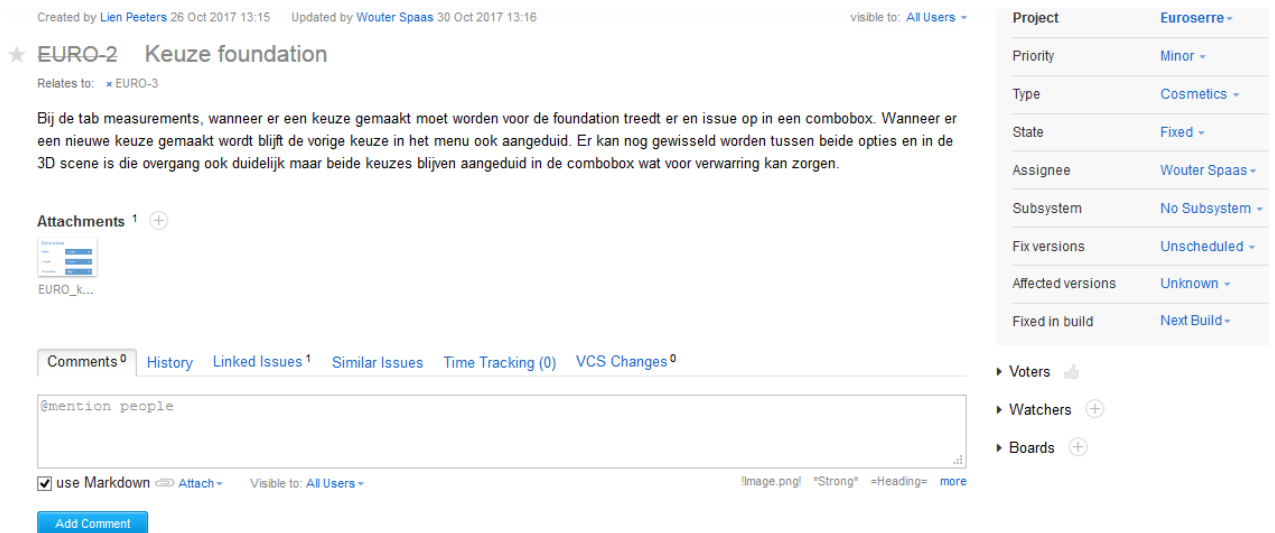
Created by [Lien Peeters](#) 26 Oct 2017 13:15 Updated by [Wouter Spaas](#) 30 Oct 2017 13:16 visible to: [All Users](#)

## ★ EURO-2 Keuze foundation

Relates to: [x EURO-3](#)

Bij de tab measurements, wanneer er een keuze gemaakt moet worden voor de foundation treedt er en issue op in een combobox. Wanneer er een nieuwe keuze gemaakt wordt blijft de vorige keuze in het menu ook aangeduid. Er kan nog gewisseld worden tussen beide opties en in de 3D scene is die overgang ook duidelijk maar beide keuzes blijven aangeduid in de combobox wat voor verwarring kan zorgen.

**Attachments** 1 [+](#)



Project	Euroserre -
Priority	Minor -
Type	Cosmetics -
State	Fixed -
Assignee	Wouter Spaas -
Subsystem	No Subsystem -
Fix versions	Unscheduled -
Affected versions	Unknown -
Fixed in build	Next Build -

Comments<sup>0</sup> [History](#) [Linked Issues](#)<sup>1</sup> [Similar Issues](#) [Time Tracking](#) (0) [VCS Changes](#)<sup>0</sup>

@mention people

use Markdown [Attach](#) Visible to: [All Users](#) [more](#)

[Add Comment](#)

[Voters](#) [Watchers](#) [Boards](#)

*Figuur 15: Voorbeeld van een issue binnen YouTrack*

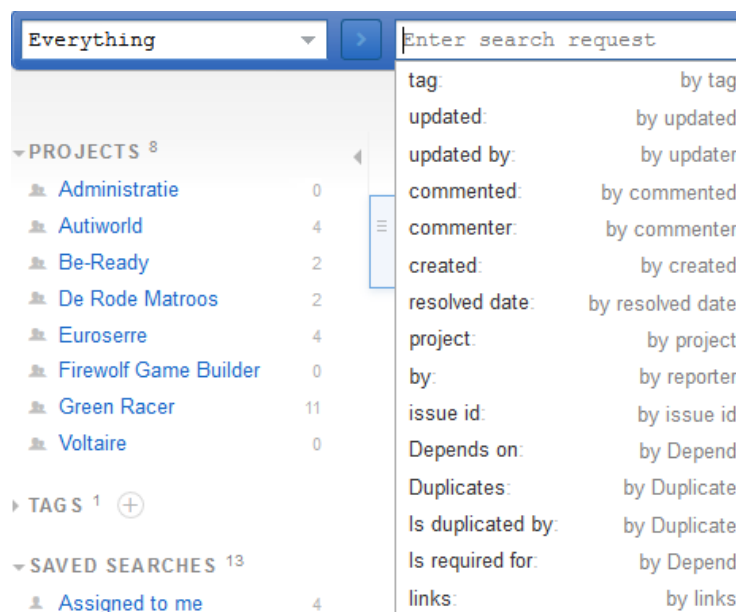
Wanneer er een Kanban project aangemaakt wordt, wordt er ook een Kanban bord aangemaakt. Deze zijn terug te vinden onder agile boards in het menu. Bij een Scrum project worden er ook agile boards gemaakt. Er wordt dan een project development en een projectmanagement bord gemaakt. Bij een default project worden er geen agile boards aangemaakt. Ook is er de mogelijkheid zelf extra agile boards aan te maken. Dit kunnen Scrum, Kanban, persoonlijke, aangepaste of versie gebaseerde borden zijn. Zo een board heeft al wat instellingen meegekregen vooraf maar net zoals de andere borden die door het systeem zelf aangemaakt worden per project, kunnen ze volledig aangepast worden. Aan een bord moet enkel steeds minimum 1 project gekoppeld zijn.

Aangemaakte issues worden vervolgens getoond in een lijst met een ID, bestaande uit de naam van het project en een volgnummer voor de issue. Ook worden er hier andere eigenschappen van de issues getoond. De issues kunnen aangepast worden door te klikken op de issue, in een lijst van issues die bestaan in het hele systeem. Figuur 16 weergeeft een fragment van de issues binnen YouTrack die ontstaan zijn tijdens de stageperiode. Er is een stuk van de figuur afgeknipt om het passend te maken op de pagina. Normaal gezien is er nog een kolom uiterst rechts waarin de naam van de gebruiker die de issue in het systeem heeft geplaatst, getoond wordt.

De gebruiker kan er voor kiezen de issues te bekijken per project. Issues kunnen ook aangepast worden door te zoeken naar de issues met de zoekbalk. Hierin geeft YouTrack enkele opties om te zoeken, dit is zichtbaar in figuur 17. Er kan ook bepaald worden waar deze zoekterm moet uitgevoerd worden links van de zoekbalk. In de plaats van de standaard 'everything' kan hier gekozen worden om bijvoorbeeld in een specifiek project te gaan zoeken. Verder kan een issue ook van status veranderen wanneer het van plaats wordt veranderd op een agile board binnen het systeem.

<input type="checkbox"/>	★	DRMA-2	Item kan niet aangeklikt worden	Normal	Bug	Not a bug	Unassigned	No Subsys...	Unscheduled	Unknown	?	?
<input type="checkbox"/>	★	DRMA-1	Fouten bij opdrachten	Normal	Bug	Can't Repr...	Unassigned	No Subsys...	Unscheduled	Unknown	?	?
<input type="checkbox"/>	★	EURO-4	Front, back, right tekst	Minor	Cosmetics	Submitted	Unassigned	No Subsystem	Unscheduled	Unknown		
<input type="checkbox"/>	★	EURO-3	Vervolg keuze glitch	Normal	Bug	Submitted	Unassigned	No Subsystem	Unscheduled	Unknown		
<input type="checkbox"/>	★	EURO-2	Keuze foundation	Minor	Cosmetics	Submitted	Unassigned	No Subsystem	Unscheduled	Unknown		
<input type="checkbox"/>	★	GR-4	Gravitypoint aanpassen in niet VR mode	Normal	Feature	Submitted	Unassigned	No Subsystem	Unscheduled	Unknown		
<input type="checkbox"/>	★	EURO-1	Text glitch	Normal	Cosmetics	Submitted	Unassigned	No Subsystem	Unscheduled	Unknown		
<input type="checkbox"/>	★	GR-1	Database velden	Normal	Feature	In Progress	Frank Van Geirt	Back-end	Unscheduled	Unknown		
<input type="checkbox"/>	★	GR-2	Google Play store: new screenshots	Normal	Task	Submitted	Wouter Spaas	No Subsystem	Unscheduled	Unknown		
<input type="checkbox"/>	★	BR-1	Connection Error	Normal	Feature	To be tested	Wouter Sp...	No Subsys...	Unscheduled	Unknown	○ 1d	?
<input type="checkbox"/>	★	BR-2	Inlog issues	Show-sto...	Bug	To be tested	Wouter Sp...	No Subsys...	Unscheduled	Unknown	?	?

Figuur 16: Fragment uit de lijst van issues binnen YouTrack



Figuur 17: Zoeken naar issues binnen YouTrack

Verder heeft YouTrack ook een dashboard. Wanneer een gebruiker zijn YouTrack systeem bezoekt is dit het eerste wat getoond wordt. Hierin kunnen standaard issues gezien worden die net geüpdatet zijn of men kan dit aanpassen naar bijvoorbeeld issues die aangewezen zijn aan die gebruiker. Op het dashboard kunnen ook *widgets* geplaatst worden. Dit kunnen rapporten zijn of notities. Met rapporten in YouTrack wordt er bijvoorbeeld een *Gantt chart*, een *burndown chart*, een chart waarin het aantal issues per project getoond wordt, ... bedoeld. YouTrack biedt hier veel opties in.

Binnen de instellingen van YouTrack kan er aan *access management* gedaan worden. Er kan bepaald worden welke gebruiker toegang heeft tot wat, welke gebruiker welke permissies heeft. Gebruikers zijn onderverdeeld aan de hand van groepen en rollen die aangepast kunnen worden naar eigen noden.

YouTrack biedt ook de mogelijkheid om *workflows* aan te passen binnen het systeem. Er kan dus aangepast worden op welke manier een issue door het systeem kan vloeien op vlak van statussen. Men kan bijvoorbeeld bepalen dat een issue van status open naar in progress mag gaan en van in progress naar done. Maar hierna niet meer naar in progress. Dit kan bijvoorbeeld toegepast worden indien het bedrijf beslist dat er in het geval dat een issue die opgelost zou moeten zijn en toch opnieuw voorkomt, er een volledige nieuwe issue aangemaakt moet worden.

Verder in de settings is de mogelijkheid om de data van het systeem te exporteren te vinden. Het bestand dat hier verkregen wordt kan door een ander YouTrack systeem opnieuw uitgelezen worden. Issues kunnen per project geëxporteerd worden naar CSV of HTML bestanden.

#### 4.2.1.4 Implementatie

YouTrack is het eerste systeem dat binnen het bedrijf is geïmplementeerd. Aangezien dit soort systeem volledig nieuw was voor het bedrijf was het dus ook een eerste leeromgeving.

Er zijn een aantal projecten gemaakt binnen het systeem, deze volgden steeds de default layout. Over Scrum werd er op dat moment namelijk nog niet gepraat binnen het bedrijf en een eerste Kanban bord werd al in gebruik genomen op een fysieke manier. De projectmanagement functies zijn daardoor niet ten volle tot hun uiting kunnen komen, maar de gebruikers hebben er wel mee geëxperimenteerd.

### 4.2.1.3 Beoordeling op basis van de aanwezigheid van de requirements

Requirement	Aanwezig	Verklaring
<i>Mobiele app</i>	<b>Ja</b>	De tool heeft een mobiele app beschikbaar voor Android en iOS systemen.
<i>Tijdschrijven</i>	<b>Ja</b>	De tool heeft de mogelijkheid bij te houden hoe lang we schatten dat een issue nodig heeft om opgelost te worden en het heeft de mogelijkheid om effectief werk op een issue te loggen. Deze <i>feature</i> werd uitgebreid in die nieuwe versie die uitkwam tijdens de loop van de stageperiode.
<i>Import/Export</i>	<b>Ja</b>	YouTrack biedt de mogelijkheid om hun data te exporteren naar CSV bestanden en HTML, en om Jira back-ups te importeren.
<i>Beschikbaarheid cloud</i>	<b>Ja</b>	De tool is beschikbaar op de cloud, maar men kan er ook voor kiezen het zelf te hosten op een eigen server. De functionaliteiten van beide producten blijft hetzelfde.
<i>Aantal projecten</i>	<b>Ja</b>	Er kan een ongelimiteerd aantal projecten aangemaakt worden binnen de tool. Op het gratis plan heeft het systeem wel een maximum opslagruimte van 5.12Gb. Dit zou geen effect mogen hebben op het aantal projecten dat in het systeem past, de ruimte wordt wel snel ingevuld door bijlages zoals screenshots toe te voegen aan issues. Projecten binnen het gratis plan zijn ook niet privé.
<i>Verskil tussen bug en feature duidelijk</i>	<b>Ja</b>	Het verschil tussen een <i>bug</i> en een <i>feature</i> is duidelijk aangegeven. De tool biedt meerdere keuzes binnen de soorten tickets die aangemaakt kunnen worden en er kunnen zelf nog types toegevoegd worden naar wens.
<i>Projectmanagement</i>	<b>Ja</b>	Binnen de tool kan men gebruik maken van agile borden. Projecten kunnen van verschillende types zijn zoals Scrum en Kanban wat er voor zorgt dat de projecten bij aanmaak over een aantal specifieke eigenschappen beschikken.
<i>Git koppeling</i>	<b>Onvolledig</b>	De tool biedt koppelingen met <i>Git</i> aan, en ook met de <i>Git</i> naar keuze van het bedrijf (Gitlab). Alleen ontstonden er toch problemen bij het proberen koppelen van de <i>Git repository</i> van het bedrijf en is de koppeling niet gelukt.
<i>IDE koppelingen</i>	<b>Onvolledig</b>	YouTrack biedt <i>plug-ins</i> voor de IDE's in de suite van JetBrains. Issues kunnen dan gezien en aangepast worden vanuit de IDE. Ook timetracking kan dan vanuit de IDE. JetBrains biedt geen support voor YouTrack <i>plug-ins</i> in andere IDE's maar er bestaan wel enkele open source <i>plug-ins</i> . Zo een <i>plug-in</i> is er voor Eclipse en voor Visual Studio Code, maar niet voor de Visual Studio versie die het bedrijf gebruikt (Visual Studio Community 2015).

Tabel 10: Aanwezigheid requirements YouTrack

#### 4.2.1.4 Feedback eindgebruikers

NR	1.1	1.2	1.3	1.4	2.1	2.2	2.3	2.4	3.1	3.2	3.3	3.4	4.1	4.2	4.3	4.4	TOT	
1.	-2	-2	-2	-2	/	/	/	/	+1	-1	0	0	0	-1	-1	0	-10	
2.	+2	/	/	/	/	/	/	/	+1	+1	+1	+1	0	-1	-1	-1	3	
NR	5.1	5.2	5.3	5.4	5.5	6.1	6.2	6.3	7.1	7.2	7.3	7.4	7.5	8.1	8.2	8.3		
1.	+2	0	0	+1	0	+1	0	0	+2	-1	-2	-1	-1	-1	-1	0	-1	
2.	+1	0	0	+1	/	/	/	/	+1	0	0	0	0	-1	0	+1	3	
NR	8.4	8.5	8.6	8.7														
1.	0	-1	-2	+1														-2
2.	+1	0	+1	+1														3
																		-4

Tabel 11: Feedback eindgebruikers YouTrack

Gebruiker 1 gaf de tool een score van -13 en gebruiker 2 gaf de tool een score van 9. De gemiddelde eindscore is dus -2, en de tool haalt een totaalscore van -4.

De tool behaalt een zeer zwakke score, ondanks de hoge plaatsing in de eerste ranglijst. Om deze score nader te kunnen verklaren bekijken we op welke categorieën de tool net zo slecht scoort. Zo kunnen we zien op welke vlakken de tool niet goed genoeg is.

	Cat. 1	Cat. 2	Cat. 3	Cat. 4	Cat. 5	Cat. 6	Cat. 7	Cat. 8
YouTrack	-6	0	5	-5	5	1	-2	1

Tabel 12: Scores per categorie YouTrack

<b>Cat. 1</b>	<i>Git</i>	Deze categorie scoort natuurlijk laag aangezien de koppeling tussen de tool en de <i>git repository</i> van het bedrijf niet lukte.
<b>Cat. 2</b>	<i>Verbindingen met IDE's</i>	Voor deze vragen waren er geen antwoorden waardoor de categorie neutraal scoort. Dit is te verklaren doordat de <i>plugins</i> voor de IDE's van deze tool niet gebruikt werden.
<b>Cat. 3</b>	<i>Access management</i>	Deze categorie scoort positief wat betekent dat het bepalen van rollen en permissies voor gebruikers vlot ging in deze tool.
<b>Cat. 4</b>	<i>Tijdschrijven</i>	De tool scoort hierop laag. De functionaliteit wordt wel aangeboden door de tool maar ze blijkt niet gebruiksvriendelijk.
<b>Cat. 5</b>	<i>Vershil tussen bug en feature duidelijk</i>	De tool scoort hier hoog op wat normaal is aangezien er veel mogelijkheden zijn tot personaliseren van de verschillende velden.
<b>Cat. 6</b>	<i>Projectmanagement</i>	Op projectmanagement scoort de tool neutraal tot positief, het krijgt in totaal 1 pluspunt. Dit kan verklaard worden door het feit dat er veel opties waren voor project management, maar dat ze niet volledig gebruikt zijn door de gebruikers.
<b>Cat. 7</b>	<i>Mobiele app</i>	De mobiele app scoort laag, de app wordt niet gebruiksvriendelijk bevonden.
<b>Cat. 8</b>	<i>Algemeen gebruik</i>	Over het algemeen zijn de eindgebruikers neutraal tot positief over de gebruiksvriendelijkheid en duidelijkheid van de tool.

Tabel 13: Toelichting scores per categorie YouTrack

#### 4.2.1.5 Conclusie

YouTrack was de eerste projectmanagement en bugtrackingtool die gebruikt werd door het bedrijf. Hierdoor waren veel concepten nieuw en moesten de gebruikers nog wennen aan het rapporteren van issues met zo een systeem. Het concept van agile boards was ook nog maar net geïntroduceerd tijdens het begin van de stageperiode door middel van een Kanban bord waardoor vooral dit onderdeel van de tool niet helemaal tot zijn recht kon komen. Ook doordat het de eerste tool was en er dus veel tijd verloren ging aan het in de routine komen van zo een tool te gebruiken, scoren er enkele categorie laag. Dit omdat functionaliteiten zoals het gebruik van de tool in vaak gebruikte IDE's hierdoor niet gebeurde. De tool zou iets hoger gescoord kunnen hebben indien het niet als eerste gebruikt werd.

### 4.2.2 JIRA

#### 4.2.2.1 Algemene informatie

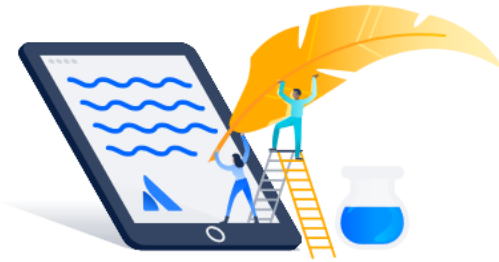
Jira is de projectmanagement en issuetracking tool van het bedrijf Atlassian. Dit bedrijf staat ook bekend voor de ontwikkeling van Trello, BitBucket en SourceTree. Ook hun tool is beschikbaar in de cloud of op een eigen server.

De focus van Jira ligt op het bieden van een tool waarin alles kan. Ze zijn voor het gebruik van één tool in de plaats van meerdere. Binnen Jira kunnen user stories, issues en sprints gepland worden en kunnen taken getraceerd worden. Er zijn rapporten beschikbaar over de data in het systeem en *workflows* kunnen aangepast worden. [13]

#### 4.2.2.2 Installatie en configuratie

Om een cloud versie van Jira op te zetten moet de gebruiker een naam voor zijn website kiezen en zich registreren met een naam, e-mail en paswoord. Vervolgens wordt er aan de gebruiker gevraagd om een mail te bekijken die de gebruiker dan zou moeten hebben ontvangen. Wanneer de link in de mail gevolgd wordt krijgen we de melding dat de website opgestart wordt. Wanneer deze opgezet is ontvangt de gebruiker hier nog eens een mail over. Wanneer de website geladen is wordt de gebruiker gevraagd om een taal in te stellen en een avatar te kiezen voor zijn account.

Vergezeld van een welkomstbericht krijgt de gebruiker de keuze om een demo project te zien, een nieuw project aan te maken of data van een andere tool in te laden. Vervolgens wordt er een nieuw project aangemaakt aan de hand van een aantal vragen, of kan er gekozen worden voor een specifiek soort project. Wanneer we de vragenlijst invullen krijgen we achteraf te zien welk soort project er bij dit soort team zou passen en kan er een naam gekozen worden voor het project. Het soort project dat aangeraden wordt is ofwel Scrum ofwel Kanban. Figuur 18 toont deze vragenlijst met antwoord en figuur 19 laat zien welke keuzes er zijn voor het aanmaken van een nieuw project zonder de vragenlijst te gebruiken.



## How does your team work?

My team **has experience with** agile methodologies.

We spend our time working on **fixing bugs**.

We have a **tight** schedule to finish our work.

Advanced

Back

Submit

Back

Create project

Your team works best with



### Scrum

In a Scrum project, you have a board and a backlog for your team to work in sprints.

Project name

E.g: "My Scrum project"

Figuur 18: Vragenlijst voor een soort project in Jira en antwoord hierop

## Create project

SOFTWARE



### Scrum software development

Agile development with a board, sprints and stories. Connects with source and build tools.



### Kanban software development

Optimise development flow with a board. Connects with source and build tools.



### Bug tracking

(Formerly Basic software development) Track dev tasks and bugs with a basic list.



### Agility

Get started on your project with this easy, simplified board.

BUSINESS



### Project management

Plan, track, and report on team projects.



### Task management

Organize the ad hoc and business-as-usual tasks for the team.



### Process management

Control the activities for specific work processes.



### Content Management

Manage business content and creative copy, from draft through to final publish.



### Recruitment

Track new hires from interview to offer.



### Document Approval

Manage the review and approval of contracts, policies, and other critical documents.



### Lead Tracking

Track sales leads and opportunities through to conversion.



### Procurement

Simplify the purchase process, from request to approval to delivery.

Figuur 19: Mogelijke soorten projecten binnen Jira



Jira valt op door het verschil in software en business projecten. Ze bieden niet alleen de populaire agile projecten Kanban en Scrum aan maar bieden ook ondersteuning voor business specifieke projecten. Dit kan dan gaan over processen zoals aanwerving, managen van allerhanden taken, lead verzameling, inkopen, ...

Stel dat we een Scrum project aanmaken, dan begint Jira door ons de backlog te tonen voor het project en kunnen we deze gaan invullen. Na het aanmaken van een project kunnen de backlog, sprints, reporten, releases, issues, ... hiervan gezien worden en kunnen deze ook aangepast worden. Standaard is een issue in een backlog ofwel een story, een taak of een *bug*. Indien er in het algemeen een issue aangemaakt wordt binnen het systeem dan kan dit ook een epic zijn.

Een nieuwe issue die in de backlog wordt aangemaakt kent enkel een titel, er worden geen andere velden ingevuld. Deze kunnen wel nadien nog ingevuld worden. Indien er in het algemeen een issue aangemaakt wordt binnen het systeem voor dit Scrum project dan zijn er meer velden beschikbaar voor de issue zoals een uitgebreide uitleg, prioriteit, labels, bijlagen, gelinkte issues, verantwoordelijke voor de issue en een volgnummer voor een sprint. Bij een Kanban project zijn dit dezelfde velden, alleen ontbreekt het veld voor een sprint nummer. Een issue 'in het algemeen' aanmaken betekent dat er een issue wordt aangemaakt binnen het hoofdmenu, en dat er tijdens het invullen van de eigenschappen pas een project wordt gekozen. Eigenschappen van issues en *workflows* van projecten kunnen naar eigen voorkeur worden aangepast.

Figuur 20 toont een voorbeeld van een issue in Jira. Het volledige scherm dat een gebruiker te zien krijgt past niet op de figuur. De verschillende opties voor een issue en eigenschappen zijn zichtbaar, rechts ontbreekt nog de aangewezen persoon die verantwoordelijk is voor de issue, de gelogde tijd op de issue en een aanmaakdatum en eventuele deadline. Dit is een epic die in progress is en waaraan 2 taken vasthangen. Onderaan kan ook de activiteit van verschillende gebruikers in verband met de issue gezien worden.

Jira werkt met agile borden en deze zijn beschikbaar voor zowel software als business projecten. Bij Kanban projecten heet het dan een *Kanban board* en bij businessprojecten gewoon een *board*. Voor een *lead tracking* board bijvoorbeeld, past het bord zich aan aan het doel van zo een project. Het bord ken dan kolommen zoals *opportunity* (opportuniteit), *contacted* (gecontacteerd), *in negotiation* (in overleg), *won* (gewonnen) en *loss* (verloren). Leads kunnen dan een correcte plaats krijgen op het bord naargelang hun status.

Buiten het aanmaken van projecten en issues biedt de tool ook een dashboard. Dit kan aangepast worden naar eigen voorkeur aan de hand van *widgets*. Er kan bijvoorbeeld getoond worden wat de laatste nieuwe issues zijn, welke issues die aan de huidige gebruiker zijn toegewezen er nieuw zijn en grafieken en tabellen over data uit het systeem.

De tool biedt ook instellingen op vlak van *access management*. Gebruikers en hun eigenschappen (zoals permissies) kunnen gezien en aangepast worden. De gebruikers kunnen ook onderverdeeld worden in gebruikersgroepen, en van deze groepen kunnen de permissies ook aangepast worden.



Voitaire2 / VOL2-3

## Tower defence gameplay

[Edit](#) [Comment](#) [Assign](#) [Backlog](#) [Selected for Development](#) [Workflow](#) [Admin](#)

### Details

Type:	<span>+</span> Epic	Status:	<span>IN PROGRESS</span> <a href="#">(View workflow)</a>
Priority:	<span>↑</span> Medium	Resolution:	Unresolved
Labels:	None		
Epic Name:	Tower defence		

### Description

De gameplay van de towerdefence volledig bouwen, met defences, offences en interactie met het weer.

### Attachments



### 2 Issues in this epic

<span>✓</span> VOL2-1	Verschillende defences toevoegen.	<span>✓</span> <span>DONE</span>	Wouter Spaas
<span>✓</span> VOL2-2	Offences toevoegen	<span>✓</span> <span>DONE</span>	Wouter Spaas

### Activity

[All](#) [Comments](#) [Work log](#) [History](#) [Activity](#)

27 November



Wouter Spaas updated 2 fields of VOL2-3 - Tower defence gameplay

- Changed the Remaining Estimate to '0 minutes'
- Logged '4 hours'

+ 1 week ago

Figuur 20: Een issue in Jira

### 4.2.2.3 Implementatie

Bij het gebruik van deze tool klonk agile projectmanagement al wat bekender in de oren voor het bedrijf dan voordien waardoor het systeem ook ten volle gebruikt werd. Het fysieke Kanban bord werd na de implementatie van deze tool niet meer gebruikt, in de plaats daarvan gebruikte met de borden die Jira beschikbaar stelde.

Binnen Jira heeft het bedrijf de Kanban projecten uitgetest als softwareprojecten. Scrum projecten binnen de tool zijn niet aan bod gekomen. De businessprojecten die getest zijn geweest zijn *test management* en *lead tracking*.

#### 4.2.2.4 Beoordeling op basis van de aanwezigheid van de requirements

Requirement	Aanwezig	Verklaring
<i>Mobiele app</i>	Ja	De tool heeft een mobiele app beschikbaar voor Android en iOS systemen.
<i>Tijdschrijven</i>	Ja	De tijd die er nodig is om een issue of taak op te lossen kan ingeschat worden en toegevoegd worden aan de issue of taak. Ook kan er een deadline aan de taak of issue gekoppeld worden en kan er worden bijgehouden hoeveel tijd er reeds aan de issue of taak is gewerkt.
<i>Import/Export</i>	Ged.	De data binnen Jira kan geëxporteerd worden als back-up. We verkrijgende de data dan in XML formaat, wat minder handig is in gebruik van .CSV bestanden. Importeren kan aan de hand van .CSV bestanden, zo is de data van YouTrack (de vorige gebruikte tool) dan ook geïmporteerd in deze tool. Exporteren naar een andere tool is dus omslachtig.
<i>Beschikbaarheid cloud</i>	Ja	De tool is beschikbaar op de cloud, maar men kan er ook voor kiezen het zelf te hosten op een eigen server. De functionaliteiten van beide producten blijft hetzelfde.
<i>Aantal projecten</i>	Ja	Er kan een ongelimiteerd aantal projecten aangemaakt worden binnen de tool.
<i>Vershil tussen bug en feature duidelijk</i>	Ja	Het verschil tussen een <i>bug</i> en een <i>feature</i> is duidelijk aangegeven. De tool biedt meerdere keuzes binnen de soorten tickets die aangemaakt kunnen worden en er kunnen zelf nog types worden toegevoegd.
<i>Projectmanagement</i>	Ja	Binnen de tool kunnen er projecten gemaakt worden die gebaseerd zijn op Scrum of Kanban. De Kanban projecten werken dan met een Kanban bord, en de Scrum projecten bieden de mogelijkheid tot het opstellen van een backlog, sprint backlogs en het verdelen van taken en issues in verschillende sprints.
<i>Git koppeling</i>	Ja	Het is mogelijk de tool te koppelen aan <i>Git</i> , en ook aan de <i>Git</i> omgeving die het bedrijf het meeste gebruikt: GitLab. Wanneer een werknemer in GitLab een nieuwe change commit, dan wordt dit in Jira vermeld samen met een link naar de commit in GitLab.
<i>IDE koppelingen</i>	Ged.	Er is een <i>plug-in</i> voor Visual Studio beschikbaar op de marketplace van Jira. Voor Eclipse worden er geen <i>plug-ins</i> meer ondersteund door Jira.

Tabel 14: Aanwezigheid requirements Jira

#### 4.2.2.5 Feedback eindgebruikers

NR	1.1	1.2	1.3	1.4	2.1	2.2	2.3	2.4	3.1	3.2	3.3	3.4	4.1	4.2	4.3	4.4	TOT	
1.	0	0	-1	+1	+2	0	0	+1	+1	-1	+1	-2	+1	+1	+1	0	5	
2.	+1	0	0	0	0	+1	0	0	+2	+2	+2	+2	+1	0	+1	+1	13	
NR	5.1	5.2	5.3	5.4	5.5	6.1	6.2	6.3	7.1	7.2	7.3	7.4	7.5	8.1	8.2	8.3		
1.	+1	+1	-1	+1	+1	+1	+1	+1	+2	+1	+1	+1	+1	+1	+1	+1	15	
2.	+2	+2	+2	+2	+2	+1	+1	+1	+1	+1	+1	0	0	+1	+2	+2	21	
NR	8.4	8.5	8.6	8.7														
1.	0	+1	+1	+1														3
2.	+1	+2	+1	+2														6
																		63

Tabel 15: Feedback eindgebruikers Jira

Gebruiker 1 gaf de tool een score van 23 en gebruiker 2 gaf de tool een score van 40. De gemiddelde eindscore is dus 31,5, en de tool haalt een totaalscore van 63.

Deze tool scoort veel hoger dan de vorige geïmplementeerde tool. Hieronder een overzicht op welke vlakken de tool uitblonk.

	Cat. 1	Cat. 2	Cat. 3	Cat. 4	Cat. 5	Cat. 6	Cat. 7	Cat. 8
Jira	2	4	8	6	13	6	9	17

Tabel 16: Scores per categorie Jira

<b>Cat. 1</b>	<i>Git</i>	De verbinding met <i>Git</i> werkte, toch krijgt het omwille van de ietwat omslachtige configuratie ervan een neutraal tot positieve score.
<b>Cat. 2</b>	<i>Verbindingen met IDE's</i>	De verbinding met IDE's was aanwezig voor Visual Studio en werkte vlot waardoor de gebruikers hierop de tool positief lieten scoren.
<b>Cat. 3</b>	<i>Access management</i>	<i>Access management</i> is zeker aanwezig binnen Jira en wordt toegepast door middel van rollen en groepen.
<b>Cat. 4</b>	<i>Tijdschrijven</i>	Tijd kan ingeschat worden voor taken, issues kunnen een deadline krijgen en men kan tijd loggen op een issue. Dit zorgt ervoor dat Jira een positieve score krijgt op tijdschrijven.
<b>Cat. 5</b>	<i>Vershil tussen bug en feature duidelijk</i>	Velden in Jira kunnen uitgebreid aangepast worden waardoor de tool hierop heel positief scoort.
<b>Cat. 6</b>	<i>Projectmanagement</i>	De tool biedt zoals eerder vermeld enorm veel soorten verschillende projecten aan en maakt ook een onderscheid tussen business projecten en software projecten. Hierdoor scoort de tool positief op deze vragen.
<b>Cat. 7</b>	<i>Mobiele app</i>	Er is een mobiele app beschikbaar voor de vereiste apparaten en deze werken volgens de gebruikers naar behoren.
<b>Cat. 8</b>	<i>Algemeen gebruik</i>	De tool scoort enorm hoog op gebruiksvriendelijkheid en overzichtelijkheid.

Tabel 17: Toelichting scores per categorie Jira

#### 4.2.2.6 Conclusie

Doordat agile werken al bekender was bij het implementeren van deze tool konden de projectmanagement functies ten volle gebruikt worden. Samen hiermee werd ook het tijdschrijven grondig uitgetest. De gebruikers waren enorm onder de indruk en hadden na de tool een week in gebruik te hebben eigenlijk al beslist dat ze deze tool wilden implementeren na het afsluiten van het onderzoek. Met deze gedachte in het achterhoofd is er dan ook getracht veel dingen te configureren zoals *access management* en *workflows* waardoor ze voor het invullen van de vragenlijst hier een duidelijker beeld van hadden.

De hoge score is evenredig aan de interesse van de gebruikers voor het product. Ook al hebben ze nog maar twee producten getest, zijn ze sterk overtuigd van dit systeem.

### 4.2.3 SpiraTeam

#### 4.2.3.1 Algemene informatie

SpiraTeam is een tool ontwikkeld door Inflectra. Inflectra biedt ook andere tools aan zoals TaraVault. Het biedt ook tools aan zoals Rapise, SpiraTest en RemoteLaunch. SpiraTeam noemen zij geen projectmanagement en bugtrackingtool maar een ALM of application lifecycle management tool. Op vlak van functionaliteiten maakt dit geen verschil voor het bedrijf. Zij kunnen door de functionaliteiten van de tool hetzelfde bereiken met deze tool als met de andere tools.

SpiraTeam is een tool dat projectmanagement en bugtracking oplossingen biedt alsook een tool is om testen te managen. Inflectra biedt ook SpiraPlan aan en SpiraTest. Deze hebben de dezelfde functionaliteiten als SpiraTeam, maar dan opgesplitst. In SpiraPlan zijn er enkel oplossingen voor projectmanagement en bugtracking en SpiraTest heeft als doel testen te managen. SpiraPlan biedt dus voor het bedrijf genoeg functionaliteiten, maar omwille van het testen binnen de stageopdracht wordt toch voor SpiraTeam gekozen.

#### 4.2.3.2 Installatie en configuratie

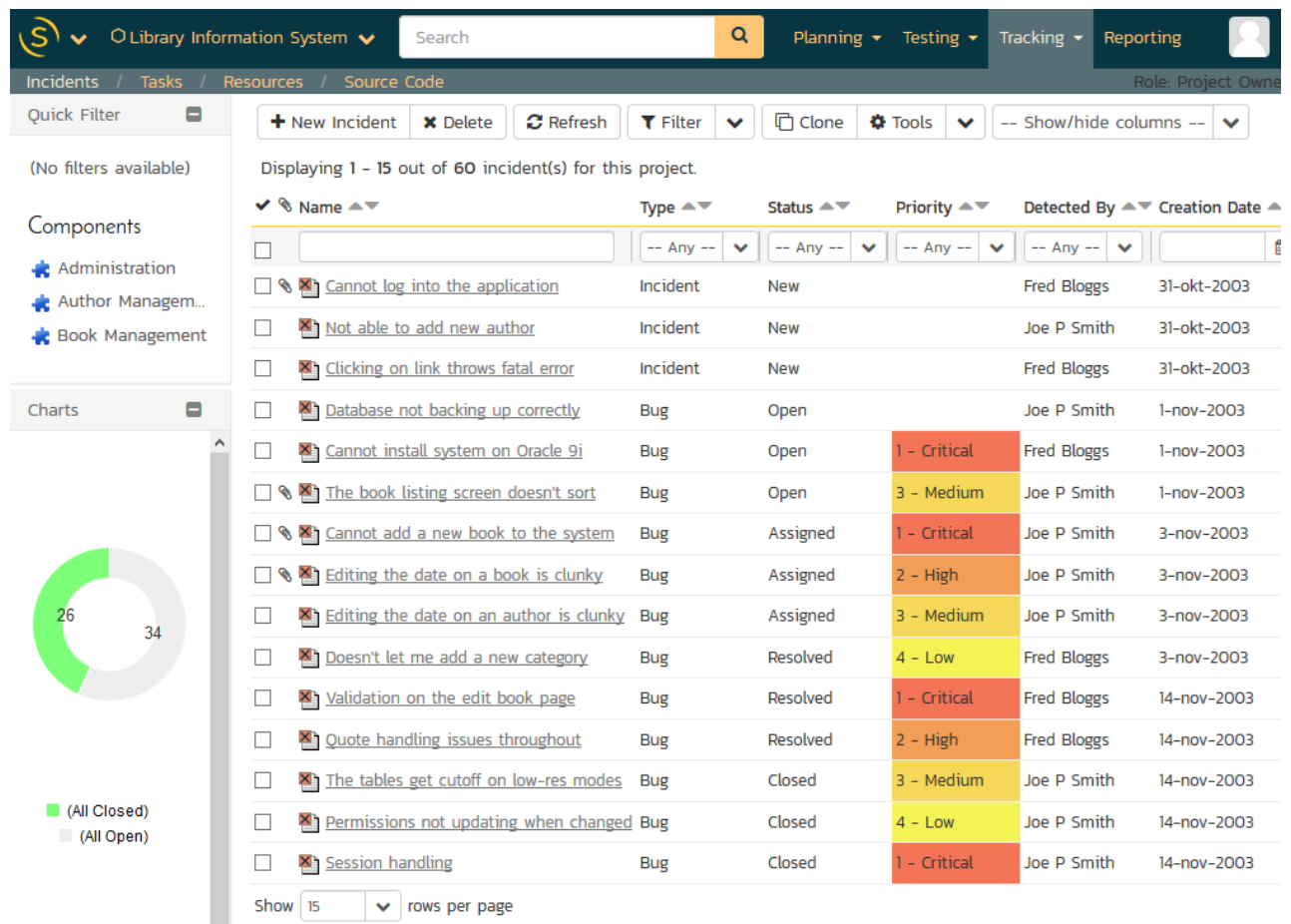
Om SpiraTeam te gebruiken, moet een gebruiker zich eerst registreren. De gebruiker kiest een product naar keuze op de website, klikt op de knop om een trial aan te vragen en maakt vervolgens zijn of haar account aan. Hierna wordt de website van de tool opgezet. De gebruiker krijgt een e-mail wanneer dit klaar is. Indien de gebruiker voor de eerste keer naar de website gaat, dan moet deze inloggen met een gebruikersnaam en paswoord die verstuurd zijn via een e-mail.

Wanneer de gebruiker inlogt wordt er een dashboard getoond met informatie uit de verschillende huidige projecten. Indien er bovenaan in het menu een specifiek project wordt gekozen, dan wordt er een dashboard getoond over dat specifieke project. Het menu bovenaan is opgedeeld in de submenu's planning, testing, tracking en reporting. Onder planning kunnen we requirements opstellen, een planning bord zien, de releases van de software in ontwikkeling bijhouden en documenten opslaan en opvragen. Onder testing hebben we verscheidene opties in verband met test management. Testcases kunnen er opgeslagen worden en de resultaten ervan kunnen hier ook worden bijgehouden.

Onder tracking vinden we incidenten, taken die moeten uitgevoerd worden, resources die beschikbaar zijn voor het project (hiermee worden de personen die meewerken aan het project bedoeld) en ook revisies van code van het project kan in het systeem worden bijgehouden. Reporting biedt uitgebreid de mogelijkheid om verschillende soorten rapporten en schema's te genereren aan de hand van data binnen het systeem.

Er kunnen twee soorten projecten aangemaakt worden, een intern of een extern project. Er is hierbij geen verschil in gebruik van de tool. Types van incidenten kunnen aangepast worden of er kunnen soorten incidenten toegevoegd worden. Er zijn standaard de soorten incidenten ingebouwd zoals bug, change request, issue, risk, training, ... De verschillende statussen die een project kan hebben en de prioriteiten kunnen ook naar wens worden aangepast. Ook de *workflows* binnen projecten kunnen naar voorkeur worden ingesteld. *Workflows* en andere settings verschillen in de categorieën planning, requirements, incidenten en testing. Voor deze 4 categorieën zijn er steeds verschillende instellingen mogelijk.

Onder de instellingen van het systeem kan er ook aan *access management* worden gedaan. Gebruikers kunnen toegevoegd worden en hun instelling kunnen aangepast worden. Ze kunnen verschillende rollen toegewezen krijgen en aan projecten worden toegevoegd.



Figuur 21: Incidenten binnen SpiraTeam

Figuur 21 toont een voorbeeld van een *interface* binnen SpiraTeam. Het toont een deel van een screenshot gemaakt binnen het menu 'incidenten' van het systeem.

### 4.2.3.3 Implementatie

SpiraTeam is veel uitgebreider product dan de vorige geïmplementeerde tools. Deze tool biedt namelijk bovenop projectmanagement en issuetracking ook de mogelijkheid om aan testmanagement te doen. De eindgebruikers hebben de demoprojecten binnen het systeem bekeken en eigen projecten proberen aanmaken maar hebben zich niet uitgebreid verdiept in het systeem. Het systeem was ingewikkeld en schrok de eindgebruikers af. Uit de demodata in het project kunnen we afleiden dat het product vooral gericht is aan grote bedrijven, wat kan verklaren waarom het product voor het stagebedrijf te uitgebreid overkwam.

### 4.2.3.4 Beoordeling op basis van de aanwezigheid van de requirements

Requirement	Aanwezig	Verklaring
<i>Mobiele app</i>	<b>Ged.</b>	De vereiste voor een mobiele app was ontstaan uit de nood naar het kunnen zien van issues en het gebruiken van de tool op verplaatsing op een mobiel apparaat. Deze tool biedt geen mobiele app, maar de website is wel aangepast naar mobiele telefoons en is responsive. De website ziet er op een smartphone uit zoals het op een app zou kunnen zijn. Vandaar dat het niet scoort als niet aanwezig.
<i>Tijdschrijven</i>	<b>Ja</b>	Tijd kan gelogd worden op de issues. Er kunnen start data en deadlines ingesteld worden.
<i>Import/Export</i>	<b>Ja</b>	De tool biedt een add-on waarmee data vanuit Excel werkbladen geïmporteerd kan worden en waarmee ook data van het systeem geëxporteerd kan worden naar Excel.
<i>Beschikbaarheid cloud</i>	<b>Ja</b>	De tool is beschikbaar op de cloud, maar men kan er ook voor kiezen het zelf te hosten op een eigen server. De functionaliteiten van beide producten blijft hetzelfde.
<i>Aantal projecten</i>	<b>Ja</b>	Er kan een ongelimiteerd aantal projecten aangemaakt worden binnen de tool.
<i>Vershil tussen bug en feature duidelijk</i>	<b>Ja</b>	Het verschil tussen <i>bugs</i> en <i>features</i> is heel duidelijk in deze tool. Standaard wordt er een onderscheid gemaakt tussen requirements, issues en taken.
<i>Projectmanagement</i>	<b>Ja</b>	De tool is ontworpen met agile in het achterhoofd. Agile borden en sprints zitten er daardoor dus standaard in verwerkt. De flow van het hele systeem is gefocust op agile werken.
<i>Git koppeling</i>	<b>Nee</b>	De tool biedt geen integratie met <i>git</i> aan.
<i>IDE koppelingen</i>	<b>Ja</b>	Voor deze tool bestaat er een <i>plug-in</i> voor Eclipse en een <i>plug-in</i> voor Visual Studio.

Tabel 18: Aanwezigheid requirements SpiraTeam



#### 4.2.3.5 Feedback gebruikers

NR	1.1	1.2	1.3	1.4	2.1	2.2	2.3	2.4	3.1	3.2	3.3	3.4	4.1	4.2	4.3	4.4	TOT
1.	/	/	/	/	/	/	/	/	0	-1	+1	-1	+1	0	0	+1	1
2.	/	/	/	/	-1	-1	0	0	+1	0	0	0	0	-1	-1	-1	-4
NR	5.1	5.2	5.3	5.4	5.5	6.1	6.2	6.3	7.1	7.2	7.3	7.4	7.5	8.1	8.2	8.3	
1.	+1	0	0	+2	+2	+1	0	+1	/	/	/	/	/	-1	0	0	7
2.	0	-1	-1	-1	-1	0	0	+1	0	0	0	0	0	+1	0	0	-2
NR	8.4	8.5	8.6	8.7													
1.	0	-1	-1	+1													
2.	0	0	-1	+1													
																	-1
																	0
																	1

Tabel 19: Feedback eindgebruikers SpiraTeam

Gebruiker 1 gaf de tool een score van 7 en gebruiker 2 gaf de tool een score van -6. De gemiddelde eindscore is dus 0,5, en de tool haalt een totaalscore van 1.

	Cat. 1	Cat. 2	Cat. 3	Cat. 4	Cat. 5	Cat. 6	Cat. 7	Cat. 8
Jira	0	-2	0	-1	1	3	0	-1

Tabel 20: Scores per categorie SpiraTeam

Cat. 1	<i>Git</i>	Aangezien er geen verbinding mogelijk was met <i>Git</i> , kon de tool niet positief scoren op dit onderdeel. De score blijft neutraal doordat de vragen niet beantwoord konden worden.
Cat. 2	<i>Verbindingen met IDE's</i>	De tool scoort slecht op de verbinding met IDE's ondanks de aanwezigheid van een <i>plug-in</i> voor beide IDE's Dit kan te wijten zijn aan het feit dat de tool slechts enkele dagen werd geïmplementeerd.
Cat. 3	<i>Access management</i>	De tool bood uitgebreide mogelijkheden aan tot <i>access management</i> maar was hier niet gebruiksvriendelijk in. Aanpassingen maken gebeurde vaak op een omslachtige manier. Desondanks scoort de tool nog neutraal.
Cat. 4	<i>Tijdschrijven</i>	Tijdschrijven was aanwezig maar niet duidelijk genoeg voor de gebruikers. Hierdoor scoort de categorie een minpunt.
Cat. 5	<i>Verskil tussen bug en feature duidelijk</i>	Verschillen tussen <i>bugs</i> , <i>issues</i> en dergelijke zijn binnen dit systeem zeker duidelijk. Ze worden in de standaard flow van het systeem zeker onderscheiden van elkaar. Op deze manier scoort de tool hier een pluspunt.
Cat. 6	<i>Projectmanagement</i>	De tool is ontwikkeld met agile in het achterhoofd wat ook duidelijk was voor de eindgebruikers. Zo scoort de tool op deze categorie het meest.
Cat. 7	<i>Mobiele app</i>	Aangezien er geen app was maar wel een responsive website was het onduidelijk of de tool nu aan deze vereiste voldoet of niet. Deze vereiste moet herzien worden.
Cat. 8	<i>Algemeen gebruik</i>	In het algemeen was de tool moeilijk in gebruik aangezien ze enorm uitgebreid was tegenover de vorige gebruikte tools en op die manier was het ook vaak onoverzichtelijk in gebruik.

Tabel 21: Toelichting scores per categorie SpiraTeam



### 4.2.3.6 Eindscore

De tool scoort laag, maar minder laag dan de eerste tool die geïmplementeerd werd. Doordat de tool niet even lang in gebruik werd genomen als de andere tools zijn niet alle functionaliteiten uitgebreid gebruikt door de gebruikers wat een effect kan hebben gehad op de scores. Het scoort ook lager omdat deze tool volgens het team te uitgebreid was, ze hadden eerder het gevoel zich extra werk te bezorgen door het configureren van deze tool dan dat het hun positieve resultaten zou leveren.

## 4.3 Vergelijking

Als deel van de vergelijking, wordt er eerst een beoordeling gemaakt op vlak van de aanwezigheid van de verschillende vereisten. 'Ged.' staat voor gedeeltelijk.

	Req 1	Req 2	Req 3	Req 4	Req 5	Req 6	Req 7	Req 8	Req 9
<i>YouTrack</i>	JA	JA	JA	JA	JA	JA	JA	Ged.	Ged.
<i>Jira</i>	JA	JA	Ged.	JA	JA	JA	JA	JA	Ged.
<i>SpiraTeam</i>	Ged.	JA	JA	JA	JA	JA	JA	NEE	JA

Tabel 22: Vergelijking aanwezigheid requirements voor alle tools

Hieronder volgt een vergelijking van alle tools op basis van hun score, samen met een gemiddelde score per tool alsook een percentage voor het aantal punten dat het heeft gehaald op het totaal.

	Gebruiker 1	%	Gebruiker 2	%	Gem.	%	Tot.
<i>YouTrack</i>	-13	-18%	9	12.5%	-2	-2.8%	-4
<i>Jira</i>	23	31.9%	40	55.6%	31.5	43.8%	63
<i>SpiraTeam</i>	7	9.7%	-6	-8.3%	0.5	0.7%	1

Tabel 23: Vergelijking scores van eindgebruikers voor alle tools

Om duidelijkheid te geven, wordt er per tool ook per categorie getoond hoeveel een tool scoort op een bepaalde categorie uit de vragenlijst. Zo kunnen we vergelijken op welk vlak een tool uitblinkt of net niet goed genoeg blijkt te zijn:

	Cat. 1	Cat. 2	Cat. 3	Cat. 4	Cat. 5	Cat. 6	Cat. 7	Cat. 8
<i>YouTrack</i>	-6	0	5	-5	5	1	-2	1
<i>Jira</i>	2	4	8	6	13	6	9	17
<i>SpiraTeam</i>	0	-2	0	-1	1	3	0	-1

Tabel 24: Vergelijking scores op basis van categorie

Tabel 25 toont nog een vergelijking tussen het aantal requirements dat de tool heeft en de score van de gebruikers.

	Aantal req. op JA	Eindscore vragenlijst
YouTrack	7	-4
Jira	7	63
SpiraTeam	7	1

Tabel 25: Vergelijking van de tools, aantal requirements versus gebruikersscore

Er was geen vereiste door het bedrijf op vlak van het maximumbudget voor deze tool. Hierbij een prijsvergelijking van de verschillende tools en hun licentieprijs voor hun cloud oplossingen.

Tool / # gebr.	1	2 - 3	4 - 5	6 - 10	11	12	13	14	15
YouTrack	/	/	/	/	/	/	/	/	€20
Jira	\$10	\$10	\$10	\$10	\$77	\$84	\$91	\$98	\$105
SpiraTeam	\$16	\$78	\$120	\$200	\$400	\$400	\$400	\$400	\$400
(SpiraPlan)	\$10	\$20	\$50	\$100	\$200	\$200	\$200	\$200	\$200

Tabel 26: Prijsvergelijking tools

## 5. Evaluatie

### 5.1 Meerwaarde onderzoek

Dit onderzoek is een uitbreiding op het onderzoek uitgevoerd tijdens de stageperiode in verband met projectmanagement binnen gamedevelopment. De implementaties hiervan worden verder gezet binnen de tool. Zo wordt er binnen deze tools een digitaal Kanban bord gebruikt in de plaats van het fysieke bord dat er eerst was. Ook voelde het bedrijf zelf de nood aan om hun issues te gaan tracken. Dit gebeurde eerst al op papier wat veel minder praktisch is en snel verloren gaat. Issues die op papier vermeld worden zijn ook onhandig wanneer er werknemers op verplaatsing zijn en nog steeds hun issues moeten kunnen lezen. Ook was het bedrijf op zoek naar een manier om aan tijdschrijven te doen, dit als een eerste stap om te bekijken hoe en waar het team productiever kan zijn en om *bottlenecks* te kunnen identificeren.

### 5.2 Bespreking resultaten

#### 5.2.1 Vergelijking met andere bronnen

Jetbrains heeft een grootschalig onderzoek uitgevoerd in verband met issuetracking tools. De onderzochte tools Jira en YouTrack komen er in voor, SpiraTeam niet. In dit onderzoek zou Jira het meest populair zijn en zouden Jira en YouTrack het beste scoren op tevredenheid van gebruikers. Jira en YouTrack zouden ook de beste functionaliteiten bieden. YouTrack wordt volgens het onderzoek door de meeste mensen aangeraden. Trello wordt aangeraden voor kleine teams. [14]

#### 5.2.2 Relevantie met stageopdracht

Het onderzoek is ontstaan uit vraag van het bedrijf zelf. Bij de aanvatting van de stageperiode is er besproken waarnaar er een onderzoek zou kunnen uitgevoerd worden. Hierbij kwam het gebruik van een projectmanagement en issuetracking tool naar boven aangezien dit raakpunten heeft met de stageopdracht. Hierin werd namelijk projectmanagement onderzocht binnen gamedevelopment.

### 5.2.3 Aanbevelingen

Het bedrijf voelt zich duidelijk goed bij Jira gezien de hoge eindscore van de gebruikers. Het heeft ook bijna volledig alle requirements die het bedrijf vereist en het is dus een geschikte tool voor het bedrijf.

In onderdeel 5.2.1 werd Trello aangekaart. Trello past ook binnen dit bedrijf. Het bedrijf heeft namelijk een aantal werknemers die geen informatica gericht werk doen en dus iets meer moeite kunnen hebben met het gebruik van dit soort tool. Trello is een veel simpelere tool dan Jira en YouTrack en zou gebruikt kunnen worden in samenwerking met Jira.

### 5.2.4 Volgende onderzoeksvraag

Indien het bedrijf na afloop van de stageperiode verder wil gaan met testen binnen het bedrijf, dan zou er gezocht kunnen worden naar een test management tool. Dit is dan een tool waarin test cases en resultaten bijgehouden kunnen worden. De functionaliteiten van SpiraTest, die ook aanwezig waren in SpiraTeam, leunen hierbij aan. Veel projectmanagement en issuetracking tools hebben hier namelijk ook *plug-ins* voor.

## 5.3 Persoonlijke reflectie

Over het algemeen is het onderzoek goed verlopen aangezien het bedrijf er voordeel uit haalde en een tool heeft gevonden dat bij hun past. De uitgebreide marktstudie heeft veel tijd in beslag genomen maar het heeft wel nut gehad om een goed beeld te krijgen op een groot aantal mogelijke oplossingen.

Bij het gebruik van de eerste tool was agile nog onbekend waardoor de tool misschien niet helemaal tot zijn recht kwam. De tool had misschien beter kunnen scoren indien agile werken al wat meer was ingeburgerd tijdens de weken ervoor. Ook was er misschien een systeem gevonden kunnen worden dat Jira kon evenaren indien er meer tijd was geweest om een tool te selecteren dat als derde getest werd. De derde tool had misschien ook beter kunnen scoren indien het evenveel dagen zou kunnen zijn gebruikt als de andere tools.

Door dit onderzoek heb ik verschillende van deze tools kort gebruikt om te kijken of ze voldeden aan de requirements van het bedrijf. Ook heb ik met drie van deze tools uitgebreide ervaring kunnen opdoen. Naar de toekomst toe zie ik het als een interessante factor dat ik binnen meerdere van deze systemen mijn weg al ken. Ik heb nu naar de toekomst toe ook in mijn achterhoofd welke systemen voor welke projecten of bedrijven geschikt zouden kunnen zijn.

## Conclusie

Tijdens het uitvoeren van de stageopdracht is er veel vrijheid geweest om onderzoek te doen naar verschillende mogelijkheden in verband met testen en projectmanagement binnen gamedevelopment. Er is onderzoek gedaan naar verschillende manieren om games te testen, welke testmethodes er voorkomen binnen gamedevelopment en op welke manieren testen binnen het stagebedrijf kunnen worden geautomatiseerd. Er is onderzocht wat Unity (de *gamedevelopmentengine* naar keuze van het bedrijf) zelf aanbiedt om games te testen, hoe er getest kan worden met Unium en Jasmine en hoe Python scripts met PyAutoGUI gebruikt kunnen worden om games te testen. De manieren van testen bleken niet allemaal feilloos. Testen met Unium staat niet toe methodes op te roepen uit de code wat het schrijven van code voor sommige testcases sterk limiteert. De informatie op vlak van het testen van games kan het bedrijf naar de toekomst toe helpen de kwaliteit van hun producten extra te ondersteunen.

In de eerste weken van de stageperiode zijn de agile methodologieën Kanban en Scrum binnen het bedrijf geïntroduceerd om het projectmanagement te optimaliseren. Hieruit is onderzoek ontstaan naar het gebruik agile methodologieën binnen gamedevelopment, en daarop volgend ook naar projectmanagement- en issuetrackingtools die het bedrijf hierbij kunnen ondersteunen. Er is onderzocht welke producten er op de markt zijn en welke voldeden aan de eisen van het bedrijf. Deze tools zijn vervolgens met elkaar vergeleken. Uit deze vergelijking zijn drie tools naar voren gekomen die het bedrijf in gebruik heeft genomen. Hun feedback over deze producten is als deel van het onderzoek geanalyseerd. Hieruit bleek dat Jira het beste binnen het bedrijf en het team past. Bijgevolg heeft het bedrijf Jira ook effectief geïmplementeerd. Deze tool kan het bedrijf naar de toekomst toe de nodige ondersteuning bieden op vlak van projectmanagement.

Kortom: de kwaliteit van de games binnen het bedrijf kan nu beter ondersteund worden door de introductie van nieuwe testmethoden en Jira, samen met de kennis die ze nu hebben op vlak van Scrum en Kanban, zal hen naar de toekomst toe helpen in de planning en opvolging van hun projecten.

## 6. Bibliografie

- [1] A. Godoy en E. F. Barbosa, „Game-Scrum: An Approach to Agile Game Development,” in *SBGames 2010*, São Carlos (SP), Brazilië, 2010.
- [2] „Game testing,” [Online]. Available: [https://en.wikipedia.org/wiki/Game\\_testing](https://en.wikipedia.org/wiki/Game_testing). [Geopend 17 oktober 2017].
- [3] A. Davis en A. Single, „What could possibly go wrong?,” 27 april 2016. [Online]. Available: <http://www.what-could-possibly-go-wrong.com/testing-for-game-development/>. [Geopend 12 december 2017].
- [4] „Unity Documentatie - Profiler,” 27 11 2017. [Online]. Available: <https://docs.unity3d.com/Manual/Profiler.html>. [Geopend 12 12 2017].
- [5] „GameBench Features,” GameBench, [Online]. Available: <https://www.gamebench.net/#features>. [Geopend 26 oktober 2017].
- [6] „Unity Documentation - Unity Test Runner,” 1 6 2017. [Online]. Available: <https://docs.unity3d.com/Manual/testing-editortestsrunner.html>. [Geopend 12 12 2017].
- [7] F. Petrillo en M. Pimenta, „Is agile out there? Agile practices in game development,” in *Proceedings of the 28th ACM International Conference on Design of Communication*, São Carlos, São Paulo, Brazilië, 2010.
- [8] M. Shilova, „Top Agile Methodologies you should know: Scrum, XP, Kanban,” Agile Code, 10 juli 2017. [Online]. Available: <https://www.agile-code.com/blog/top-agile-methodologies/>. [Geopend 8 januari 2018].
- [9] S. Aleem, L. F. Capretz en F. Ahmed, „Game development software engineering process life cycle: a systematic review,” *Journal of Software Engineering Research and Developmen*, 2016.
- [10] J. Koutonen en M. Leppänen, „How Are Agile Methods and Practices Deployed in Video Game Development? A Survey into Finnish Game Studios,” *Agile Processes in Software Engineering and Extreme Programming*, nr. 149, pp. 135-149, 2013.
- [11] „Comparison of issue-tracking systems,” Wikipedia, 14 november 2017. [Online]. Available: [https://en.wikipedia.org/wiki/Comparison\\_of\\_issue-tracking\\_systems](https://en.wikipedia.org/wiki/Comparison_of_issue-tracking_systems). [Geopend 3 oktober 2017].
- [12] „YouTrack,” JetBrains, [Online]. Available: <https://www.jetbrains.com/youtrack/>. [Geopend 9 januari 2018].
- [13] „Jira Software,” Atlassian, [Online]. Available: <https://nl.atlassian.com/software/jira>. [Geopend 11 januari 2018].
- [14] JetBrains, „Issue Tracking Tools Survey 2016,” 2016. [Online]. Available: <https://www.jetbrains.com/youtrack/promo/tracking-tools-report-2016/>. [Geopend 17 december 2017].

## **Bijlagen**

- A. Bijlage A: Voorbeelden testdocumenten**
- B. Bijlage B: Selectie uit het SRS**
- C. Bijlage C: Lijst van tools**
- D. Bijlage D: Vragenlijst tools**

## A. Voorbeelden testdocumenten

Testapproach voor game 'Voltaire':

Projectnaam: <i>Voltaire2</i>		Release	
Datum: <i>13 december 2017</i>		Versie <i>0.1</i>	

Features to be tested						
Datum	ID	FTT	Prioriteit	Autom.	Tested	Datum
13/12/2017	FTT 1,00	<i>Functionality</i>				
13/12/2017	FTT 1,01	De gebruiker verdient energie tijdens het spel	Hoog	Ja	Ja	21/12/2017
13/12/2017	FTT 1,02	De gebruiker kan defences aankopen	Hoog	Nee	Ja	21/12/2017
13/12/2017	FTT 1,03	De gebruiker kan defences plaatsen op het veld	Hoog	Nee	Ja	21/12/2017
13/12/2017	FTT 1,04	De gebruiker kan de enemy verslaan	Hoog	Nee	Ja	21/12/2017
13/12/2017	FTT 1,05	De gebruiker verliest energie wanneer hij defences aankoopt	Hoog	Nee	Ja	21/12/2017
13/12/2017	FTT 1,06	De gebruiker verliest indien de enemy zijn eindbestemming bereikt	Hoog	Nee	Ja	21/12/2017
13/12/2017	FTT 1,07	Defences kunnen gereactiveerd worden	Medium	Nee	Ja	21/12/2017
13/12/2017	FTT 1,08	De gebruiker kan de enemy damage voor middel van zijn defences	Hoog	Nee	Ja	21/12/2017
13/12/2017	FTT 1,09	Er kunnen niet meer defences geplaatst worden dan er hotspots zijn	Medium	Nee	Ja	21/12/2017
13/12/2017	FTT 1,10	Defences kunnen terug van hun hotspot verwijderd worden	Laag	Nee	Ja	21/12/2017
	FTT 2,00	<i>User experience</i>				
13/12/2017	FTT 2,01	Het spel heeft een correcte moeilijkheidsgraad voor kinderen	Medium	Nee	Nee	/
13/12/2017	FTT 2,02	Het spel heeft een correcte snelheid voor kinderen	Laag	Nee	Nee	/

Selectie uit het testdesign voor game 'Voltaire':

Projectnaam: <i>Voltaire2</i>	Release
Datum: <i>13 december 2017</i>	Versie <i>0.1</i>
FTT <b>1.04</b>	De gebruiker kan de enemy verslaan
	Hoog

Test condities									
Cond nr.	Test cases	TC_01	TC_02	TC_03	TC_04	TC_05	TC_06	TC_07	TC_08
Cond_01	De eerste HP balk wordt verslagen	T	F	F					
Cond_02	De tweede HP balk wordt verslagen	F	T	F					
Cond_03	De derde HP balk wordt verslagen	F	F	T					
Expected result									
Exp nr.	Test cases	TC_01	TC_02	TC_03	TC_04	TC_05	TC_06	TC_07	TC_08
Exp_01	De haan wordt teruggezet aan het begin van de lane	T	T	F					
Exp_02	De speler wint het spel	F	F	T					
	Pass/Fail	Pass	Pass	Pass	To do	To do	To do	To do	To do
	Issue nr.	N.v.t.	N.v.t.	N.v.t.	N.v.t.	N.v.t.	N.v.t.	N.v.t.	N.v.t.
	Execution date	21/12/2017	21/12/2017	21/12/2017					

Projectnaam: <i>Voltaire2</i>	Release
Datum: <i>14 december 2017</i>	Versie <i>0.1</i>
FTT <b>1.05</b>	De gebruiker verliest energie wanneer hij defences aankoopt
	Hoog

Test condities									
Cond nr.	Test cases	TC_01	TC_02	TC_03	TC_04	TC_05	TC_06	TC_07	TC_08
Cond_01	De gebruiker koopt een lamp	T	F	F					
Cond_02	De gebruiker koopt een gate	F	T	F					
Cond_03	De gebruiker koopt een mudpool	F	F	T					
Expected result									
Exp nr.	Test cases	TC_01	TC_02	TC_03	TC_04	TC_05	TC_06	TC_07	TC_08
Exp_01	De gebruiker verliest 10 energie	F	F	T					
Exp_02	De gebruiker verliest 30 energie	F	T	F					
Exp_03	De gebruiker verliest 40 energie	T	F	F					
	Pass/Fail	Pass	Pass	Pass	To do	To do	To do	To do	To do
	Issue nr.	N.v.t.	N.v.t.	N.v.t.	N.v.t.	N.v.t.	N.v.t.	N.v.t.	N.v.t.
	Execution date	21/12/2017	21/12/2017	21/12/2017					



## B. Selectie uit het SRS

Het SRS is te groot om volledig in de bijlage op te nemen. Hierbij volgt een selectie van enkele pagina's uit het document

## 2. Algemene beschrijving

### 2.1 Product perspectief

#### 2.1.1 Algemeen

Het systeem heeft als doel om het maken van serious games mogelijk te maken binnen bedrijven zelf, door niet technische mensen, op een simpele en gebruiksvriendelijke manier. Deze serious games kunnen dan toegepast worden om werknemers op te leiden, of om informatie te verschaffen aan klanten.

Momenteel werkt de studio volgens een B2B principe waar er games specifiek gemaakt worden voor bedrijven, via dit systeem zullen bedrijven dan hun eigen game kunnen gaan ontwikkelen binnen het bedrijf. De studio levert de assets of 'bouwblokken' voor de games aan en de bedrijven kunnen daarna zelf aan de slag hiermee.

Ook is het de bedoeling dat bedrijven hun eigen assets kunnen gaan gebruiken binnen hun games, denk hierbij aan hun eigen logo's, video's, muziek, pdf's, ... en dergelijke. Ook moet de mogelijk voorzien worden voor de studio om bepaalde extra assets aan te kunnen bieden via de website tegen betaling.

Een ander aspect van het systeem zal rond gebruikersgegevens gaan. De bedrijven zullen de mogelijkheid moeten hebben om de gegevens van de gebruikers die een game spelen te analyseren. Aan de hand van bijvoorbeeld highscores en gespeelde tijd kunnen dan analyses gemaakt worden in verband met de game en het effect dat het heeft op de gebruikers.

De medewerkers van de studio zullen ook de mogelijkheid moeten hebben tot het beheren van het systeem. Voor hun zal er de mogelijkheid moeten zijn alle bedrijven en dus ook gebruikersgroepen (verder toegelicht in het volgende hoofdstuk) te kunnen bekijken en beheren, alsook alle projecten en games die op het systeem staan. Dit vooral in verband met in een latere versie customer support mogelijk te maken.

Er zal ook een mogelijkheid moeten zijn tot het monitoren van de performance van het systeem. Afhankelijk van hoeveel gebruikers er uiteindelijk simultaan dit product zullen gebruiken zou het handig zijn een manier te hebben om de prestaties van het systeem te monitoren.



### 2.1.2 Specifiek

Het systeem zal bestaan uit twee delen, die in een latere fase van het project uitgebreid zullen kunnen worden. In een eerste fase zal het bestaan uit een web portal en een Unity based executable, beide beschikbaar op de pc als platform.

Via het web portaal zal de gebruiker zich moeten registreren en het product moeten downloaden.

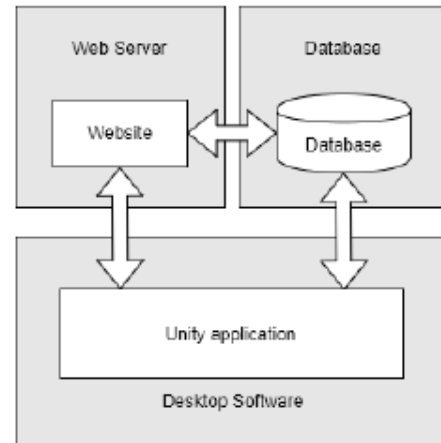
Een gebruiker die een licentie aankoopt binnen het product noemen we vervolgens de 'administrator'. Deze persoon kan vervolgens de editor gebruiken om games te maken die in de database opgeslagen worden en hij kan ook games spelen.

De administrator heeft ook de mogelijkheid om mensen uit te nodigen tot het systeem. Hij kan werknemers uitnodigen door een account te maken en deze kunnen vervolgens inloggen en de games spelen.

Het spelen van de games zorgt ervoor dat er gebruikersgegevens ontstaan en opgeslagen worden in de database. Hiervan kan de administrator dan rapporten opvragen om deze gegevens te kunnen analyseren.

Zoals eerder vermeld zijn er dus verschillende rollen van users nodig binnen het systeem. De administrator die de licentie aankoopt en de games kan editen en de gebruikersgegevens kan bekijken, en de werknemers die de games kunnen zien en spelen. Per administrator en dus bedrijf, verwachten we dat er een gebruikersgroep zal ontstaan. Alle werknemers die uitgenodigd worden komen terecht in een groep van werknemers van dat bedrijf en kunnen dan de games zien die binnen dat bedrijf gemaakt zijn, afhankelijk van de instellingen die de administrator zal maken.

Ook zal er de mogelijkheid zijn dat het hele systeem beheerd kan worden door werknemers van Firewolf Studios zelf. Zij kunnen dan alle games zien en beheren, dit als eerste stap naar customer support toe.

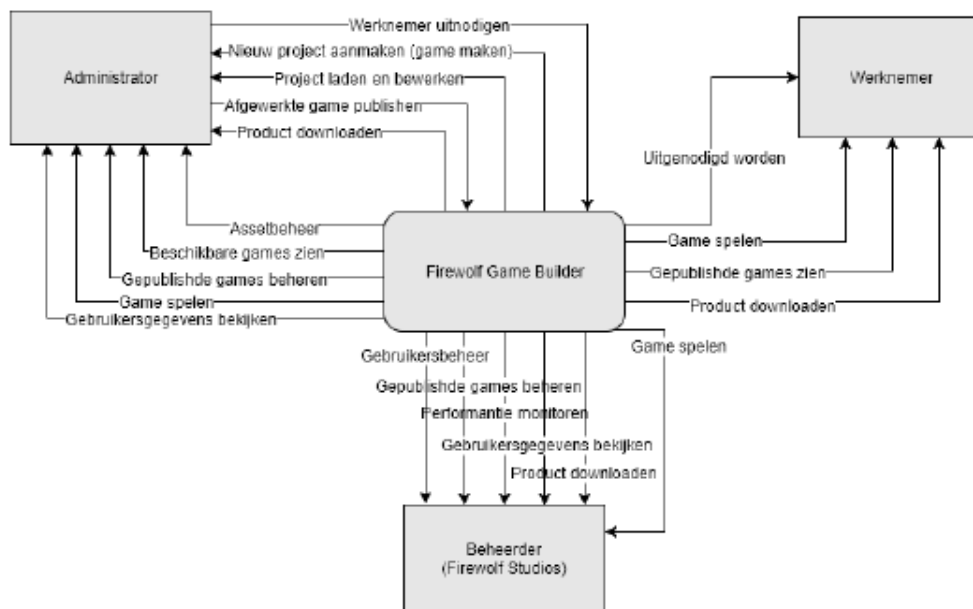


*De studenten kunnen zelf nog met ideeën op vlak van specificaties en infrastructuur naar voren komen.*



## 2.2 Productfuncties

### Context diagram



### Functies

- Registreren gebruiker
- Inloggen website gebruiker
- Uitloggen websitegebruiker
- Licenties bekijken in de webshop
- Licentie aankopen gebruiker
- Aangekochte licenties zien als administrator
  - o De gebruiker wordt een administrator als deze een licentie aankoopt
- Executable downloaden via de website
- Inloggen in de executable
- Uitloggen in de executable
- Nieuw project aanmaken
- Projecten bekijken
- Project bewerken
- Project publiceren
- Assets bekijken
- Assets toevoegen
- Assets aankopen
- Lijst van games die gepubliceerd zijn zien
- Werknemers uitnodigen door administrator
- Games uitvoeren
- Rapporten in verband met de gebruikersgegevens opvragen
- Prestatie rapporten opvragen



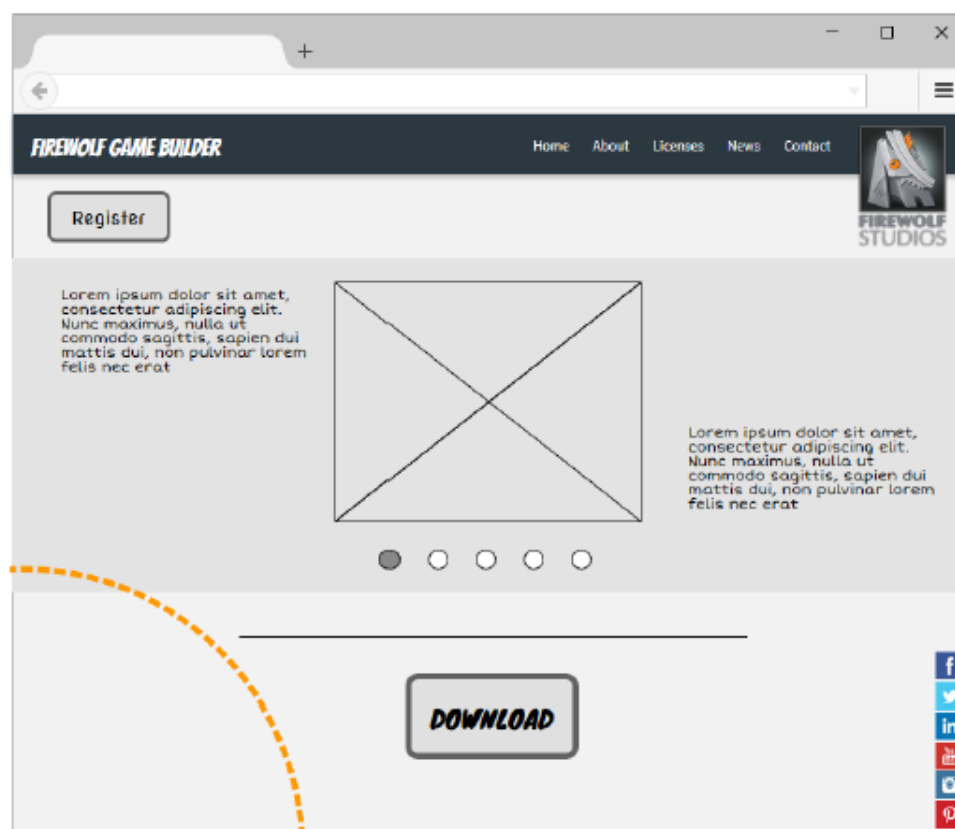
## 3. Specifieke requirements

### 3.1 Externe interface requirements

#### 3.1.1 Interfaces van de website

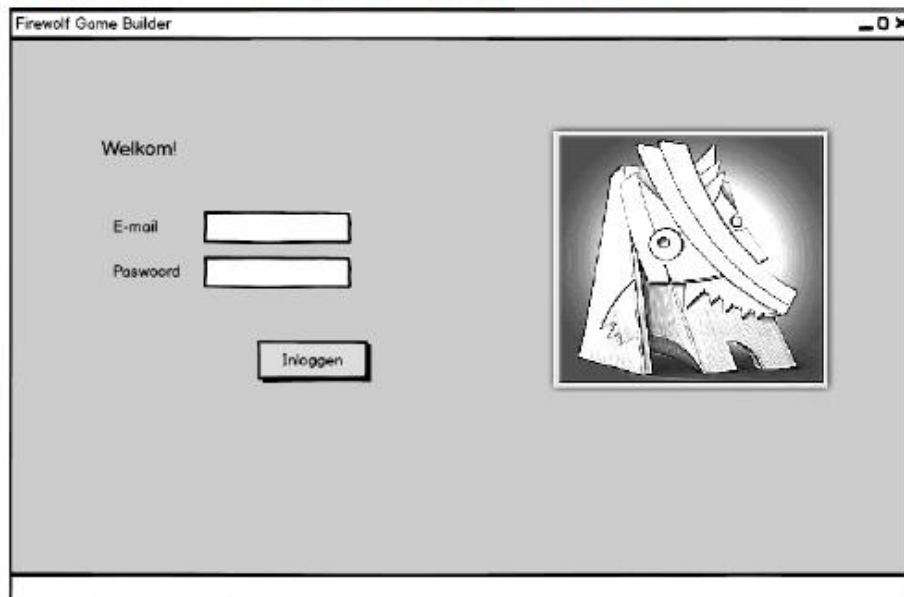
##### Administrator

Wanneer een gebruiker de website bezoekt, kan hij het product hier downloaden en kan hij zich registreren om het product te kunnen gebruiken. Het kader met het kruis door is een placeholder voor mogelijke screenshots van games die al eerder gemaakt zijn met dit product of screenshots gemaakt in de editor van het product.

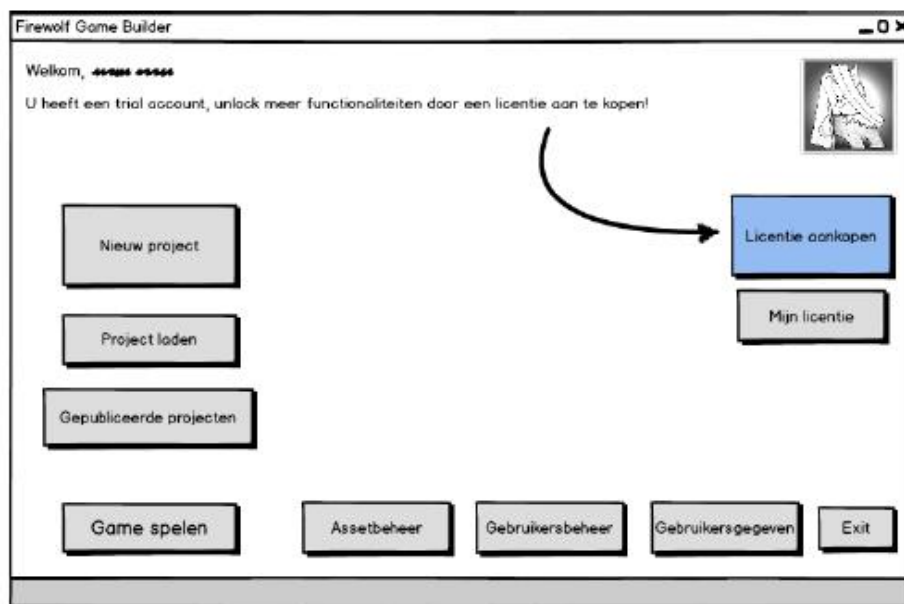


### 3.1.4 Interfaces van de Game Editor

Vooraleer de gebruiker verder kan zal hij moeten inloggen in het product

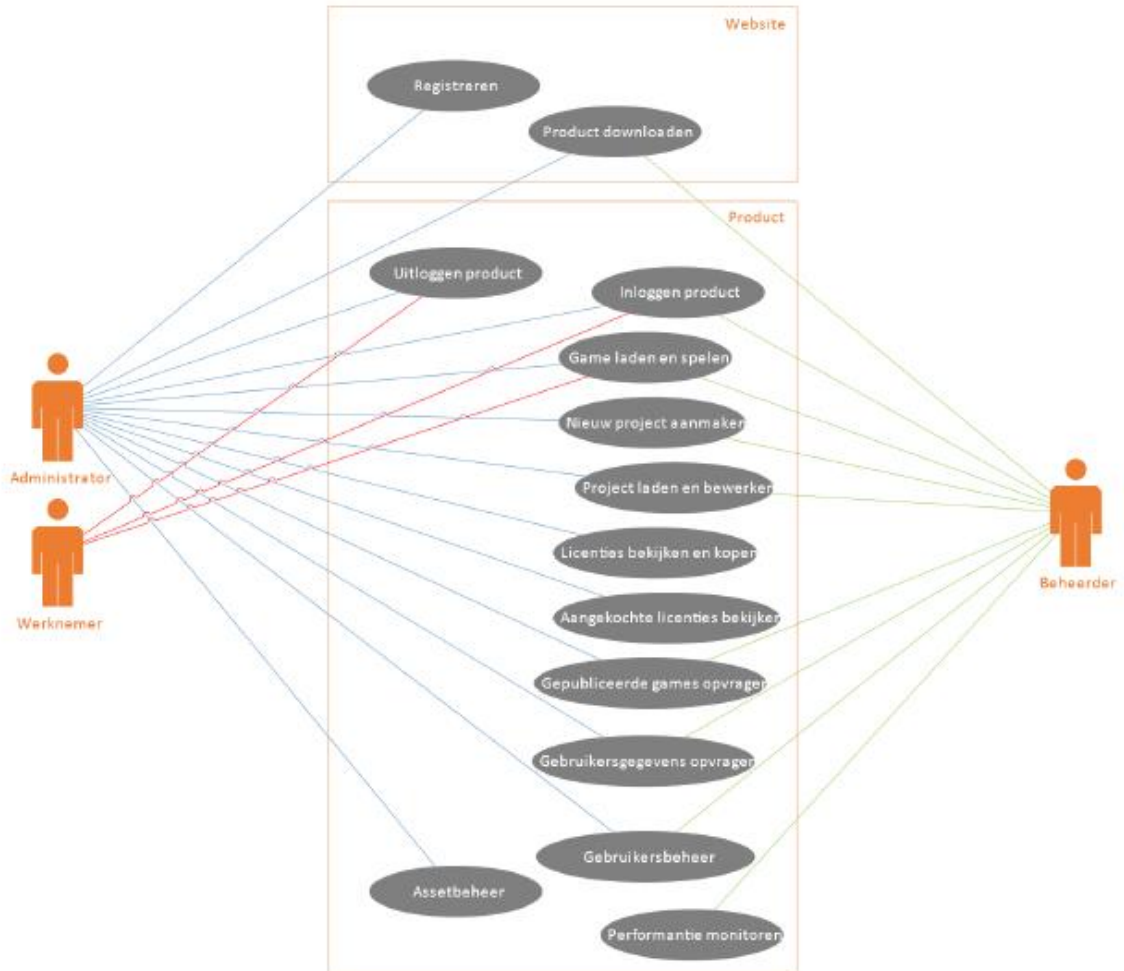


Hierna krijgt hij zijn homepage te zien. Hij kan extra content unlocken door een licentie aan te kopen



### 3.2 Functionele requirements

#### Algemene use case



### 3.2.1.1 Systeemfeature 1.1: Registreren

#### Beschrijving

Gebruikers registreren zich in het systeem zodat hun gegevens bijgehouden kunnen worden.



#### Stimulus/Antwoordvolgorde

- **Gebruiker:** Gebruiker die het product wil gaan kopen en implementeren in zijn bedrijf, administrator
- **Start:** Gebruiker navigeert naar de website
- **Precondities:** Gebruiker is verbonden met het internet
- **Normal flow:**
  1. De gebruiker klikt op de knop 'Registreren'
  2. De gebruiker vult het formulier in
  3. Het systeem controleert of het e-mailadres al in gebruik is.
  4. Het systeem slaat de gegevens op in de databank
  5. Het systeem toont de hoofdpagina opnieuw
- **Exceptions:**
  - 3.1 Het e-mailadres komt al voor in de database
  - 3.2 Het systeem geeft aan dat het e-mailadres al in gebruik is en toont het ingevulde formulier opnieuw met een lege inputfield voor het e-mailadres. Het raadt de gebruiker aan een ander e-mail adres te gebruiken omdat het huidige reeds in gebruik is.
- **Postcondities:** De gebruiker is geregistreerd in het systeem

*Eventueel kan er vereist worden dat de gebruiker zijn account verifieert door middel van een link in een e-mail*

#### Functionele requirements en dependencies

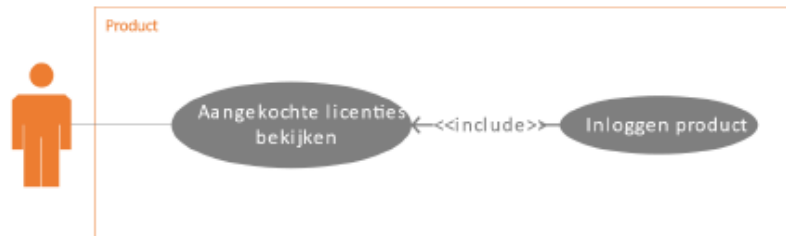
Nr.	Omschrijving	Prioriteit
1.	Registreren van een gebruiker	



### 3.2.1.6 Systeemfeature 1.6: Aangekochte licenties bekijken

#### Beschrijving

De gebruiker heeft de mogelijkheid de details van zijn licentie te bekijken, zoals de vervaldatum en wanneer hij de licentie heeft aangekocht



#### Stimulus/Antwoordvolgorde

- Gebruiker: Administrator
- Start: De gebruiker klikt op 'mijn licentie'
- Precondities: De gebruiker heeft een licentie en is momenteel ingelogd
- Normal flow:
  1. Het systeem vraagt het licentietype en de vervaldatum van deze gebruiker op
  2. Het systeem geeft de informatie over de licentie
- Postcondities: De gebruiker beschikt over de nodige informatie van zijn licentie.

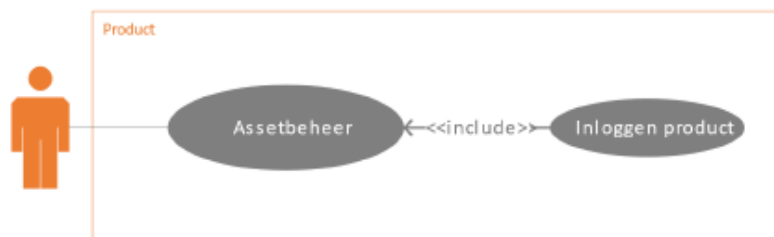
#### Functionele requirements en dependencies

Nr.	Omschrijving	Prioriteit
2.	Inloggen van een gebruiker in het product	
4.	Tonen van de licentieshop	
5.	Betalen van producten	
6.	Weergeven van informatie van aangekochte licenties	

### 3.2.1.7 Systeemfeature 1.7: Assetbeheer

#### Beschrijving

Via deze optie kan de gebruiker assets uploaden uit zijn eigen bedrijf of nieuwe bundels van assets aankopen van de studio





## Stimulus/Antwoordvolgorde

- Gebruiker: Administrator
- Start: De gebruiker klikt op de knop 'Assetbeheer'
- Precondities: De gebruiker is ingelogd
- Normal flow:
  1. De gebruiker krijgt een pagina in verband met assetbeheer te zien waar hij enkele keuzes kan maken
  2. De gebruiker klikt op de knop 'Assets uploaden'
  3. Het systeem vraagt de huidige assets van deze gebruiker op
  4. Het systeem toont in een pop-up de huidige assets van de gebruiker
  5. De gebruiker navigeert naar de plaats in de mappen structuur waar hij zijn bestand wil uploaden
  6. De gebruiker klikt op de knop 'upload'
  7. De gebruiker krijgt in zijn eigen systeem een pop-up waar hij kan navigeren naar zijn gewenste bestand en het kan selecteren.
  8. Het systeem controleert of het formaat van het bestand bruikbaar is voor de editor van het product
  9. Het systeem uploadt het vereiste bestand naar de database
  10. Het systeem toont dit bestand nu in de mappenstructuur
- Alternate flow:
  - 2.1 De gebruiker klikt op de knop 'asset shop'
  - 2.2 De gebruiker krijgt een pagina te zien met een aantal bundels aan assets die de studio beschikbaar heeft gesteld
  - 2.3 De gebruiker selecteer de gewenste asset en klikt op kopen
  - 2.4 De gebruiker wordt doorverwezen naar een pagina in het product om de gekozen asset te betalen
  - 2.5 Het systeem geeft een melding weer dat de aankoop succesvol verlopen is en dat de assets vanaf nu beschikbaar zullen zijn in de editor
  - 4.1 De gebruiker klikt op de knop 'terug'
  - 4.2 De gebruiker krijgt opnieuw de pagina in verband met assetbeheer te zien
- Exceptions:
  - 8.1 Het vereiste bestand is niet in een formaat dat binnen de editor van het product gebruikt kan worden
  - 8.2 Het systeem geeft een melding dat het bestand niet binnen de editor gebruikt kan worden omdat het geen bruikbaar formaat is binnen het systeem
  - 8.3 De gebruiker krijgt de pop-up met zijn mappenstructuur van assets te zien en kan er voor kiezen opnieuw op 'upload' te klikken en een ander bestand te selecteren
- Postcondities: De gebruiker heeft eventuele nieuwe assets beschikbaar in zijn product

*Opmerking bij exceptions rond stap 8: welke formaten van bestanden er bruikbaar zullen zijn in de editor moet duidelijk worden tijdens de ontwikkeling hiervan*

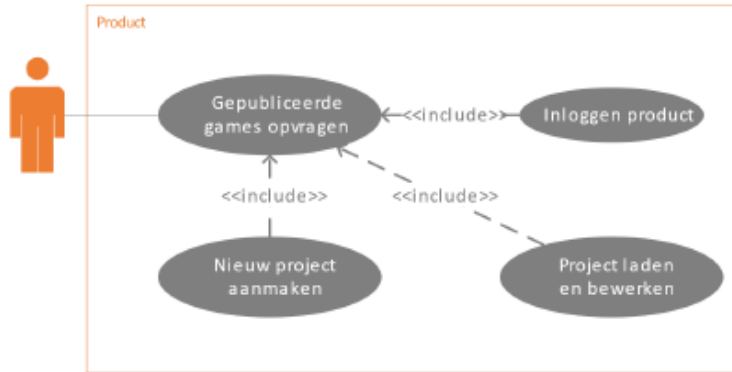
## Functionele requirements en dependencies

Nr.	Omschrijving	Prioriteit
2.	Inloggen van een gebruiker in het product	
10.	Uploaden van eigen assets van het bedrijf/de gebruiker	
11.	Tonen van de assetshop	
5.	Betalen van producten	



3.2.1.12 *Systeemfeature 1.12: Gepubliceerde games opvragen***Beschrijving**

De administrator kan projecten die hij gepubliceerd heeft zien en eventueel opnieuw verbergen.

**Stimulus/Antwoordvolgorde**

- **Gebruiker:** Administrator
- **Start:** De gebruiker klikt op 'Gepubliceerde projecten'
- **Precondities:** De gebruiker is ingelogd
- **Normal flow:**
  1. Het systeem weergeeft de projecten die voor dit bedrijf publiek weergegeven worden
  2. De gebruiker klikt eventueel op de knop 'verberg' om het project te verbergen, dit verwijdert het project niet
- **Exceptions:**
  - 1.1 Voor dit bedrijf zijn er nog geen gepubliceerde projecten
  - 1.2 Het systeem geeft een melding dat er nog geen gepubliceerde projecten zijn voor het bedrijf en een korte instructie in verband met hoe dit opgelost kan worden
- **Postcondities:** De gebruiker ziet zijn gepubliceerde projecten en heeft de mogelijkheid deze eventueel te verbergen

**Functionele requirements en dependencies**

Nr.	Omschrijving	Prioriteit
2.	Inloggen van een gebruiker	
18.	Editor openen	
19.	Project opslaan	
21.	Games publiek maken binnen het systeem	
22.	Publieke games beheren (publiceren/verbergen)	



### 3.3 Requirements tabel

Nr.	Omschrijving	Prioriteit
1	Registreren van een gebruiker	
2	Inloggen van een gebruiker	
3	Uitloggen van een gebruiker uit het product	
4	Tonen van de licentieshop	
5	Betalen van producten	
6	Weergeven van informatie van aangekochte licenties	
7	Uploaden van laatste build product naar het systeem	
8	Downloaden van het product door eindgebruiker	
9	Installeren van het product door eindgebruiker	
10	Uploaden van eigen assets van het bedrijf/de gebruiker	
11	Tonen van de assetshop	
12	Weergeven van gebruikerspermissies	
13	Aanpassen van gebruikerspermissies	
14	Onderverdelen van gebruikers in gebruikersgroepen per bedrijf	
15	Uitnodigen van een werknemer	
16	Opslaan van gebruikersgegevens tijdens spel	
17	Weergeven van gebruikersgegevens van de games	
18	Editor openen	
19	Project opslaan	
20	Project laden en verder bewerken	
21	Games publiek maken binnen het systeem	
22	Publieke games beheren (publiceren/verbergen)	
23	Game uitvoeren	
24	Monitoren van de prestaties van het systeem	



## C. Lijst van tools

Analyse tools Firewolf Studios		MUST HAVES								
Naam tool	TOT	Mobile connection	SC	Tijdschriften	SC	Import/Export data	SC	SAAS	SC	Taal / Aanpasbaar
YouTrack (JetBrains)	91	iOS en android app	10	time management	10	import en export naar versch websites:	10	Licentie	8	Java
Perforce	88	duidelijk iOS en android app	10	time management	10	niet aangegeven	10	Licentie	8	ASP.NET
FogBugz	84	FogBugz mobiele website (geen af)	10	sterk aanwezig	5	niet aangegeven	5	Licentie	8	ASP.NET
Visual Studio team services	83	Community made apps	8	niet aangegeven	5	niet aangegeven	5	Licentie	8	Java
Jira Software	83	Jira mobile	10	niet per persoon	5	Export mogelijkheid	10	Licentie	8	Java
SpiralTeam	82	iOS en android app	10	niet aangegeven	5	niet aangegeven	5	Licentie	8	Java afh van licentie
TwoProject	81	afhankelijk van licentie	5	time tracker	10	data exports en imports mogelijk	10	Licentie	8	C#
Axosoft	81	Geen app	0	time management	10	niet aangegeven	5	Licentie	8	
Plan.io	79	iOS en android app	10	time management	10	free data migration	10	Licentie	10	Ruby / JS
OpenProject	78	Geen app	0	time management	10	niet aangegeven	5	Open source/ cloud licentie	8	
Zoho projects	77	App beschikbaar iOS en Android	10	time tracker / time sheet	10	niet aangegeven	5	Licentie	8	
OTRS	76	OTRS mobile	10	time management	10	free data migration	10	SaaS licentie mogelijk	8	Perl
Plain ticket	71	Geen app, werkt op elke device	10	niet aangegeven	5	data exports mogelijk	10	Licentie	8	Java
Redmine	71	plugin iOS en android	3	aanwezig	10	Export mogelijkheid	10	Open source	0	Ruby
BugHerd	70	Geen app	0	niet aangegeven	5	niet aangegeven	5	Licentie	8	
Visual Studio team foundation	68	Geen app	0	niet aangegeven	5	niet aangegeven	5	Licentie	8	
SprintGround	67	Geen app	0	time tracking	10	niet aangegeven	5	Licentie	8	
Write	66	duidelijk iOS en android app	10	niet aangegeven	5	data imports mogelijk	7	Licentie	8	Java
Tuleap	66	Enkel android app aanwezig	7	niet aangegeven	5	data exports en imports mogelijk	10	Open source	0	PHP
PractiTest	66	Geen app	0	time management	10	niet aangegeven	5	Licentie	8	Ruby / Clojure
StarTeam	66	iOS app	7	niet aangegeven	5	niet aangegeven	5	Licentie	8	C++ / Java
DeBugMe	66	Geen app	0	time tracking	10	niet aangegeven	5	Licentie	8	eigen webpage
InstaBug	64	iOS en android app	10	time tracker	5	niet aangegeven	5	Licentie	8	NET / Java
Quality center enterprise (HP)	63	Geen app	0	tracking voor bugs	5	niet aangegeven	5	Licentie	10	PHP
Mantis bug tracker	61	Community driven apps	8	niet aangegeven	5	Data kan meegenomen na opzeg	5	Opensource / mantis hub licentie	10	PHP
PivotalTracker	61	iOS en android app	10	niet aangegeven	5	niet aangegeven	5	Licentie	8	Ruby
DevSuite (Game dev)	61	App aanwezig voor suite	10	onduidelijk	0	niet aangegeven	5	Licentie	8	C++ / JS
TechExcel DevTrack (! DevSuite hoort hierbij)	61	App aanwezig voor suite	10	onduidelijk	0	niet aangegeven	5	Licentie	8	C++ / JS
QuickBase	59	iOS en android app	10	niet aangegeven	5	niet aangegeven	5	Licentie	8	C# / Java
SysAid (ticketing system tool)	57	iOS en android app	10	timers	10	niet aangegeven	5	Licentie	8	PHP (en veel andere)
Faveo	56	iOS app / responsive website	10	time management	10	niet aangegeven	5	Licentie / open source	10	PHP
Sentry	56	duidelijk iOS en android app	10	niet aangegeven	5	niet aangegeven	5	Licentie	8	PHP
The Bug Genie	54	Geen app	0	niet aangegeven	5	niet aangegeven	5	Licentie	8	
Hansoft	53	Geen app	0	niet aangegeven	5	niet aangegeven	5	Licentie	8	
Acenre	52	Geen app	0	time tracker	10	niet aangegeven	5	Licentie	8	
Usersnap	46	Geen app	0	niet aangegeven	5	niet aangegeven	5	Licentie	8	Python / JS
Support Works	44	iOS en android app	10	niet aanwezig	0	niet aangegeven	5	Licentie	8	C++ / PHP
Bugzilla	40	iOS en android app (community)	8	time tracking	10	niet aangegeven	5	Open source	0	Perl
FusionForge	37	Geen app	0	task management	5	niet aangegeven	5	Open source	8	PHP
Sifter	36	Geen app	0	niet aanwezig	0	niet aangegeven	5	Licentie	8	
Liberum Help Desk	34	Geen app	0	time management	10	niet aangegeven	5	Open source	0	ASP.NET
Rational collaborative lifecycle management	34	Geen app	0	niet aangegeven	5	niet aangegeven	5	Licentie	8	
Launchpad suite	29	Geen app	0	niet aangegeven	5	niet aangegeven	5	Open source	0	JS / Python
zipBoard	28	Geen app	0	niet aanwezig	0	niet aangegeven	5	Licentie	8	
Trac	25	Geen app	0	niet aangegeven	5	niet aangegeven	5	Open source	0	Python
Debbugs	19	Geen app	0	niet aanwezig	0	niet aangegeven	5	Open source	0	Perl
Fossil	17	Geen app	0	niet aanwezig	0	niet aangegeven	5	Open source	0	C

Naam tool	SHOULD HAVE			MICE TO HAVES			Cloud / On premise		
	# projecten	SC Bug vs feature verschil	Project planning	Koppeling met git	Ander koppelingen	Cloud / On premise			
YoutTrack (JetBrains)	∞	10 verschil aan te duiden	9 scrum/kanban	9 GitHub, GitLab, BitBucket	10 zendesk, testrail, testlodge	5 Beide			
Perforce	∞	10 verschil duidelijk	10 agile/devops	10 git support	10 veel integration opties	5 Beide			
FogBugz	∞	10 type aan te duiden	9 agile	9 git	10 treilo en webhook	5 Cloud			
Visual Studio team services	∞	10 Features en bugs	10 scrum, kanban, backlog	9 git	10 eclipse/intellij, vis studio	8 Cloud			
Jira Software	∞	10 eventueel extra plugins	7 confluence	8 Git integratie / bitbucket	10 SourceTree	5 Beide			
SpiralTeam	∞	5 heel duidelijk plugins	10 agile	9 git	10 Visual studio / Eclipse	10 Beide			
Twproject	∞	10 verschil duidelijk	10 agile, boards, scrum	10 niet aangegeven	5 niet aangegeven	3 Beide			
Axosoft	∞	10 type aan te duiden	9 scrum, kanban, workflow	10 gitlab, github, bitbucket	10 visual studio, zendesk, ...	10 Beide			
Plan.io	3-7 (max 40/100)	5 'tasks'	8 agile	7 Giten subversion	10 redmine	3 Cloud			
OpenProject	∞	10 verschil duidelijk	10 agile/scrum	9 git	10 plugins	4 Beide			
Zoho projects	1 gratis 20 betaald	4 heel duidelijk verschil	10 veel opties	10 niet aangegeven	5 dropdown integratie	7 Cloud			
OTRS	∞	10 verschil duidelijk	10 niet aanwezig	0 niet aangegeven	5 plugins	3 Beide			
Plain ticket	∞	10 user defined	9 kanban, gant, veel opties	5 niet aangegeven	5 niet aangegeven	1 Cloud			
Reclaim	∞	10 verschil aan te duiden	10 kanban	10 git plugin	10 verschillende plugins	4 Vooral zelf			
BugHerd	∞	10 verschil duidelijk	10 scrum, kanban, backlog	10 git	10 redmine, Jira	5 Cloud			
Visual Studio team foundation	∞	10 Features en bugs	10 scrum, kanban, burndown	10 niet aangegeven	10 eclipse/intellij, vis studio	10 On premise			
SprintGround	∞	10 verschil duidelijk	9 planning, gantt	9 geen koppeling	5 niet aangegeven	1 Cloud			
Write	∞	10 user defined	10 agile, kanban, scrum	10 git integratie	0 geen koppeling	4 On premise			
Tuleap	∞	10 verschil duidelijk	10 agile	9 niet aangegeven	10 Jenkins	4 Cloud			
PractiTest	∞	10 onduidelijk	3 niet aangegeven	9 niet aangegeven	5 niet aangegeven	1 Cloud			
StarTeam	∞	10 onduidelijk	3 kanban	9 niet aangegeven	5 visual studio, eclipse	8 Cloud			
DeBugMe	5	4 niet aangegeven	3 kanban	9 git, bitbucket, ...	10 visual studio	9 Cloud			
InstaBug	2	2 niet aangegeven	3 niet aanwezig	5 git	10 heel veel andere integraties	3 Cloud			
Quality center enterprise (HP)	∞	10 verschil duidelijk	10 niet aanwezig	0 onduidelijk	5 veel IDE's	10 Beide			
Mantis bug tracker	∞	5 vooral bugs	6 scrum plugin	7 geen koppeling	0 geen koppeling	0 Beide			
PivotalTracker	10	5 labels	5 planning	9 niet aangegeven	5 niet aangegeven	1 Cloud			
DevSuite (Game dev)	∞	10 Features	5 planning visueel	9 niet aangegeven	5 niet aangegeven	1 Cloud			
TechnExcel DevTrack (! DevSuite hoort hierbij)	∞	10 Features	5 agile en gantt	9 niet aangegeven	5 niet aangegeven	1 Cloud			
QuickBase	niet aangegeven	5 niet aangegeven	3 niet aangegeven	5 niet aangegeven	5 salesforce, zendesk + eigen	5 Cloud			
SysAid (ticketing system tool)	niet aangegeven	5 tickets	3 niet aanwezig	0 niet aangegeven	5 niet aangegeven	1 Beide			
Favo	∞	10 ticketing system	7 niet aanwezig	0 geen koppeling	0 veel plugins, geen IDE plugins	4 On premise			
Sentry	∞ (gelim, events)	5 niet aangegeven	3 niet aangegeven	5 git	10 veel proj managemen. tools	3 Onduidelijk, trial aanvr			
The Bug Genie	1 (of meer betalend)	1 custom issue fields	4 agile	9 Source control integrated	10 geen koppeling	0 Beide			
Hansoft	niet aangegeven	5 heel duidelijk verschil	10 kanban, gant, scrum	9 git integratie	10 niet aangegeven	1 On premise			
Acentre	niet aangegeven	5 onduidelijk	3 project tracker	7 niet aangegeven	5 niet aangegeven	1 Cloud			
Usersnap	3	3 verschil aangegeven	10 geen, enkel bug track	0 niet aangegeven	5 integrations met andere prod	2 Cloud			
Support Works	niet aangegeven	5 Incident management	0 niet aanwezig	0 niet aangegeven	5 niet aangegeven	10 Beide			
Bugzilla	niet aangegeven	5 bugs, results in veel format	10 niet aanwezig	0 geen koppeling	0 eigen extensies schrijven mog	2 On premise			
FusionForge	niet aangegeven	5 onduidelijk	3 niet aanwezig	0 geen koppeling	0 plugins	3 Cloud			
Sifter	10	5 verschil duidelijk	3 niet aanwezig	0 geen koppeling	0 geen koppeling	8 Cloud			
Liberum Help Desk	niet aangegeven	5 onduidelijk	10 niet aangegeven	5 niet aangegeven	5 niet aangegeven	1 On premise			
Rational collaborative lifecycle management	niet aangegeven	5 niet aangegeven	3 niet aanwezig	0 geen koppeling	0 geen koppeling	0 Cloud			
Launchpad suite	niet aangegeven	5 onduidelijk	3 niet aangegeven	5 niet aangegeven	5 niet aangegeven	1 On premise			
zipBoard	1-5 (max 20)	4 niet aangegeven	3 niet aanwezig	0 geen koppeling	0 geen koppeling	0 Cloud			
Trac	niet aangegeven	5 verschil duidelijk	10 niet aanwezig	0 geen koppeling	0 geen koppeling	0 On premise			
Debbugs	niet aangegeven	5 type aan te duiden	9 niet aanwezig	0 geen koppeling	0 geen koppeling	0 On premise			
Fossil	niet aangegeven	5 onduidelijk	3 niet aanwezig	0 vs git	0 geen koppeling	0 On premise, geen server			

## D. Vragenlijst tools



### Evaluatieformulier bug tracking- en projectmanagementtools Firewolf Studio

Algemeen

Naam tool: \_\_\_\_\_

Naam participant: \_\_\_\_\_

Functionality		Zeer Oneens	Oneens	Neutraal	Eens	Zeer Eens
Type of nr.	Vraagstelling <i>Extra info en tips over vraag</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Interoperability	De tool biedt een verbinding met Git. <i>Indien ja, dan gelden volgende vragen:</i>	<input type="checkbox"/> ja <input type="checkbox"/> nee				
1.1	De verbinding tussen de tool en Git biedt een nuttige meerwaarde aan het gebruik van de tool.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.2	De verbinding tussen de tool en Git werkt vlot.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.3	De verbinding tussen de tool en Git werkt probleemloos.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.4	De verbinding tussen de tool en Git is overzichtelijk.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Interoperability	De tool biedt verbindingen met enkele IDE's. <i>Indien ja, dan gelden volgende vragen:</i>	<input type="checkbox"/> ja <input type="checkbox"/> nee				
2.1	De tool biedt een verbinding met de IDE's die ik vaak gebruik.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.2	De tool biedt verbindingen met IDE's binnen de noden van het bedrijf.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.3	De verbinding met de IDE is nuttig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.4	De verbinding met de IDE is overzichtelijk. <i>Hier gaat het vooral over de layout.</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Security	De tool biedt usermanagement, verschillende rollen en permissies. <i>Indien ja, dan gelden volgende vragen:</i>	<input type="checkbox"/> ja <input type="checkbox"/> nee				
3.1	De usermanagement functies binnen de tool voldoen aan de noden van het bedrijf.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3.2	De usermanagement opties en permissies zijn overzichtelijk.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3.3	De usermanagement opties en permissies zijn duidelijk.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3.4	Het aanpassen van permissies en rollen verloopt vlot.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

1





<b>Functionality - Continued</b>		<b>Zeer Oneens</b>	<b>Oneens</b>	<b>Neutraal</b>	<b>Eens</b>	<b>Zeer Eens</b>
Functionality	De tool biedt een mogelijkheid tot tijdschrijven. <i>Indien ja, dan gelden volgende vragen:</i>	<input type="checkbox"/> ja <input type="checkbox"/> nee				
4.1	De functies voor het tijdschrijven die de tool biedt voldoen aan de eisen van het bedrijf.	○	○	○	○	○
4.2	Het tijdschrijven binnen de tool is duidelijk.	○	○	○	○	○
4.3	Het tijdschrijven binnen de tool is overzichtelijk.	○	○	○	○	○
4.4	Het tijdschrijven binnen de tool verloopt vlot.	○	○	○	○	○
Functionality	Bepaalde velden binnen de tool kunnen aangepast worden. <i>Het is bijvoorbeeld mogelijk om verschillende soorten issues aan te maken (bug, feature, ...)</i> <i>Indien ja, dan gelden volgende vragen:</i>	<input type="checkbox"/> ja <input type="checkbox"/> nee				
5.1	Het type van een issue kan volledig naar eigen wensen aangepast worden.	○	○	○	○	○
5.2	Het aanpassen van een veld is niet moeilijk en verloopt vlot.	○	○	○	○	○
5.3	Het is duidelijk en overzichtelijk hoe ik een veld moet aanpassen.	○	○	○	○	○
5.4	Buiten het type van een issue, zijn er genoeg velden aanpasbaar.	○	○	○	○	○
5.5	De velden die ik aanpassen wil kan ik volledig naar mijn eigen wensen aanpassen.	○	○	○	○	○
Functionality	De tool biedt een projectmanagement functie. <i>Indien ja, dan gelden volgende vragen:</i>	<input type="checkbox"/> ja <input type="checkbox"/> nee				
6.1	De projectmanagement functie biedt een meerwaarde aan de tool.	○	○	○	○	○
6.2	De projectmanagement functie voldoet aan de noden van het bedrijf.	○	○	○	○	○
6.3	De projectmanagement functie is overzichtelijk.	○	○	○	○	○
<b>Portability</b>		<b>Zeer Oneens</b>	<b>Oneens</b>	<b>Neutraal</b>	<b>Eens</b>	<b>Zeer Eens</b>
Adaptability	De tool biedt een mobiele app <i>Indien ja, dan gelden volgende vragen:</i>	<input type="checkbox"/> ja <input type="checkbox"/> nee				
7.1	De app functioneert op mijn meest gebruikte mobiele toestel en OS.	○	○	○	○	○
7.2	Ik kan op de app even overzichtelijk de issues bekijken als op de desktop versie.	○	○	○	○	○
7.3	De layout en UI van de app zijn aantrekkelijk en overzichtelijk.	○	○	○	○	○
7.4	Navigatie binnen de app verloopt vlot.	○	○	○	○	○
7.5	Ik kan even makkelijk issues in de app aanpassen als op de desktop versie.	○	○	○	○	○

2



Algemeen – Usability – Efficiency		Zeer Oneens	Oneens	Neutraal	Eens	Zeer Eens
Attractiveness 8.1	De tool is in het algemeen overzichtelijk. <i>De UI is duidelijk.</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Understandability 8.2	In het algemeen is het makkelijk om in de tool te navigeren naar de gewenste onderdelen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Attractiveness 8.3	Ik heb het gevoel dat het gebruik van deze tool nuttig is. <i>Het werk gaat bijvoorbeeld sneller vooruit, of ik heb het gevoel dat ik productiever ben.</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Attractiveness 8.4	Ik vind dat mijn werk overzichtelijker is nu ik deze tool gebruik.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Understandability 8.5	Ik vind deze tool gebruiksvriendelijk.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Understandability / Learnability 8.6	Ik vind dit een gebruiksvriendelijke tool die iedereen binnen het bedrijf zal kunnen begrijpen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Time behaviour 8.7	Ik vind dat de tool vlot genoeg reageert, ik moet nooit te lang wachten voor het reageert.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>



