



Debrief

Mobile game project

Inhoudsopgave

Opdracht

Vacatures

- User Experience Designer (MV + II)
- Story Designer (IS + MV)
- Game Developer (CT)
- Gameplay Developer (CT + II)
- Projectleiders

Competenties

- Onderzoeken en begrijpen
- Conceptualiseren
- Verbeelden en prototypes maken
- Evaluieren
- Intern -en multidisciplinair samenwerken
- Initiëren, organiseren en regisseren

Werkwijze

- Scrum afspraken
- Github
- Google Docs/Drive/Hangout
- Benodigdheden

Communicatie

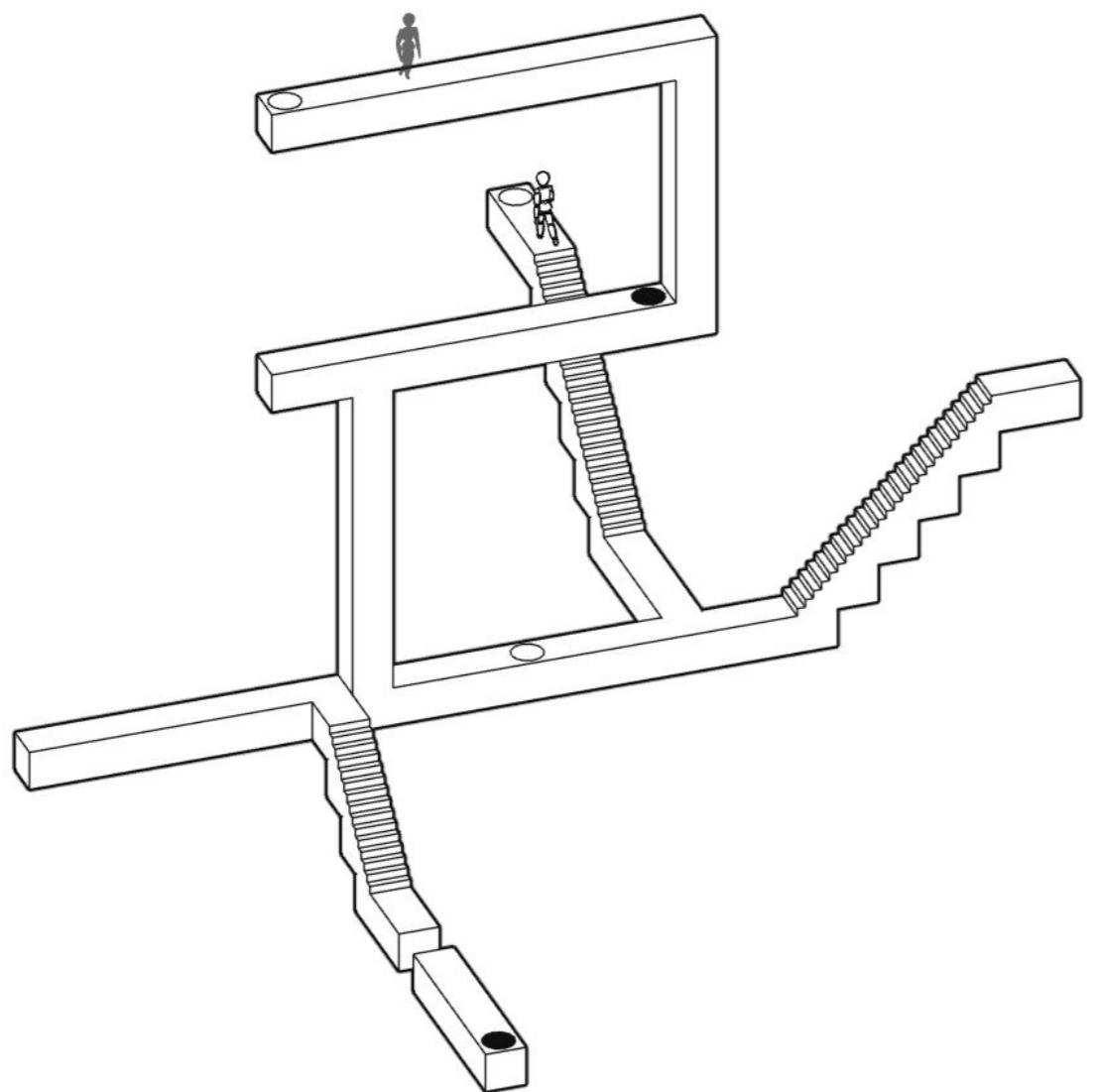
- Intern
- WhatsApp
- Google+ Community
- Google Hangout
- JIRA
- Extern
- Avans email
- Blog

Kosten

Verwachtingen van projectleden

Derde partijen

Rechten



Opdracht

Wij gaan een game ontwikkelen voor mobile devices. Dit gaan wij doen voor de volgende besturingssystemen: iOS en Android waarbij wij andere besturingssystemen en devices bij voorhand niet willen uitsluiten. Om een goed beeld van de mogelijkheden te krijgen willen wij hier een onderzoek naar doen. Uit dit onderzoek moet blijken welke technieken wij het beste kunnen gebruiken gelet op de product- en proceseisen.

Bij het maken van het concept en de interactie gaan wij letten op de volgende producteisen. De game moet een unieke gameplay hebben

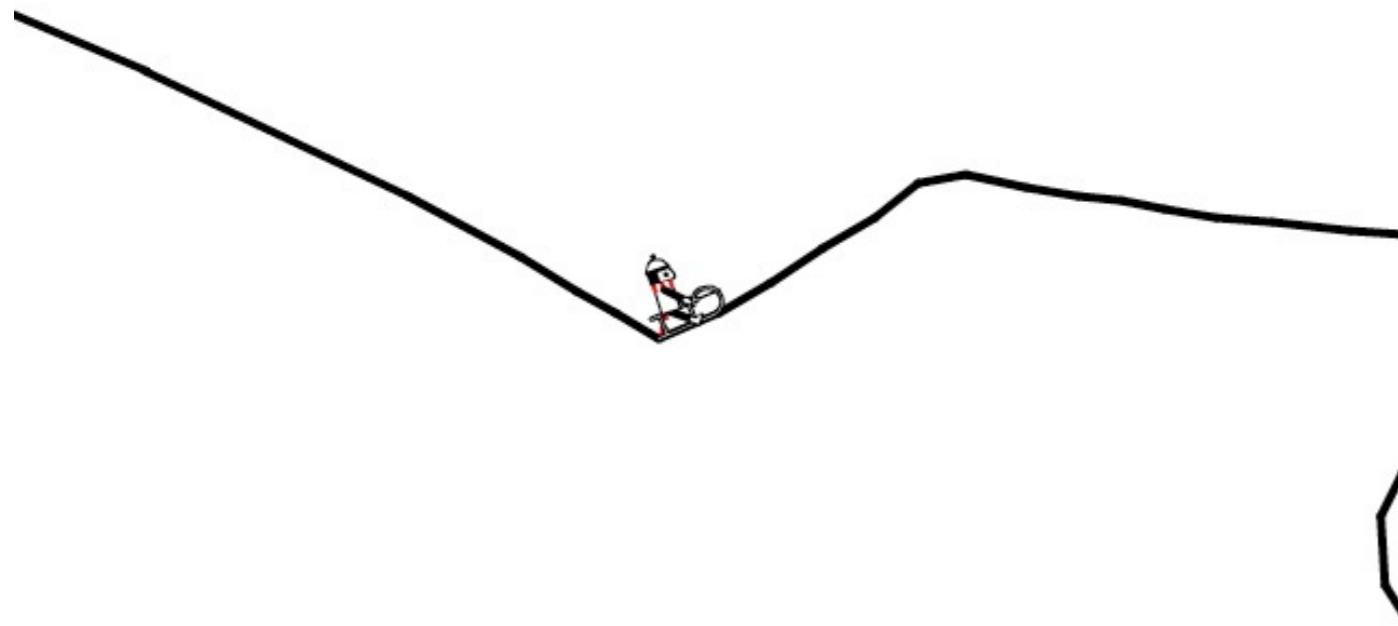
door middel van een verbinding met real-life. bijvoorbeeld motion sensors of fysieke verbinding met een medespeler. In het spel moeten puzzel elementen voorkomen en in het spel moet een beloningssysteem aanwezig zijn. Het spel moet ruimte bieden voor de speler om de omgeving van het spel te ontdekken.

De omgeving moet voldoen aan enkele vormgevingseisen. Zo moet het spel in een vrolijke stijl vormgegeven worden. Deze stijl moet tevens abstract zijn. 3D wordt niet uitgesloten maar op het moment wordt uitgegaan van 2D vanwege de haalbaarheid.

Afbeelding 1. Echochrome (JAPANstudio)

Vacatures

Doordat Scrum wordt gebruikt als ontwikkelmethode kunnen de vacatures met elkaar overlappen. Er is altijd de mogelijkheid dat je andere taken moet/kan oppakken dan in de functieomschrijving is opgesteld. Functies zijn er voor een onbepaalde tijd. Het kan voorkomen dat door procesverloop er nieuwe functies bijkomen, veranderen of verdwijnen.



Afbeelding 2. Line Rider (Boštjan Čadež)

User Experience Designer (MV + II)

Ben jij visueel ingesteld en heb je een affectie voor interactie? In deze functie krijg je de kans om je zelf verder te ontwikkelen als UX Designer. Je gaat je richten op het visueel bedenken en uitwerken de gameplay. Dit houd in dat je bi-

#UXD, #UCD, #interaction, #appearances, #illustrations, #animation, #research, #vectors, #UID

jvoorbeld schetsen, tekeningen, illustraties en animaties maakt. Hierbij zet je de ervaring van de gebruiker centraal want jij weet wat de gebruiker wilt door jouw onderzoek.

Story Designer (IS + MV)

Een goed story design is meer dan alleen maar een goed verhaal. Jij gaat je bezig houden met alle facetten van story design. Jij bent de bedenker van de gamewereld, je schrijft het verhaallijn van het spel tot in de details. Zo help je mee met

het bedenken van de achtergrond van de wereld maar ook over de eigenschappen en achtergrondgegevens van eventuele karakters in het spel. Het is een pre als je het story design ook visueel kunt uitwerken in schetsen en game art.

#storytelling, #emotional design, #characters, #research, #communication, #narrative, #entertainment, #monomyth, #game art

Game Developer (CT)

Ben jij visueel ingesteld en heb je een affectie voor interactie? In deze functie krijg je de kans om je zelf verder te ontwikkelen als UX Designer. Je gaat je richten op het visueel bedenken en uitwerken de gameplay. Dit houd in dat je bi-

#coding, #backend, #frontend, #API, #OOP, #KISS, #DRY, #frameworks

jvoorbeld schetsen, tekeningen, illustraties en animaties maakt. Hierbij zet je de ervaring van de gebruiker centraal want jij weet wat de gebruiker wilt door jouw onderzoek.

Gameplay Developer (CT + II)

Een game werk pas echt goed als de gameplay ook goed uitgewerkt is. Jij gaat je bezig houden met het maken van prototypes en iteraties waarmee je gameplay mechanisms gaan per-

#coding, #interaction, #gameplay, #UID, #testing, #prototypes, #rapid development

fectioneren. Door deze rapid development kun je de interactie goed testen en de techniek verder uit werken en optimaliseren.

Projectleiders

De projectleiders van dit project zijn Davey van der Woert en Jesse van Rheenen. De functie van een projectleider is om het project in goede banen te leiden. De projectleiders zijn tevens de Scrum Masters. Projectleiders maken besluiten,

hakken knopen door en stellen deadlines. Daarnaast springen projectleiders ook in bij andere taken waar dat nodig is. Bijvoorbeeld bij het programmeren of het uitwerken van de interactie.

Competenties

zie ook bijlage 2 en 3 voor een volledig uitgewerkte verdeling van de competenties.

Onderzoeken en begrijpen:

in de eerste fase van het project zal een onderzoek gedaan worden. Elk projectlid zal hier aan meedoen vanuit zijn eigen taak en invalshoek.

Conceptualiseren:

Voor de game moeten er concepten gemaakt geïteneerd op bedachte concepten. Afhankelijk van de taak van de projectlid wordt hier in meer of mindere mate aan geparticeerd.

Verbeelden en prototypes maken:

Er worden op verschillende vlakken prototypes gemaakt. Zo moet er visueel geprototyped worden maar ook met de achterliggende techniek.

Evaluieren:

Na elke sprint in Scrum wordt er geëvalueerd. Wat zijn 'Go's' en wat zijn 'No's'. Ook na onderzoeken, gebruikerstesten en dergelijken wordt er geëvalueerd. Niet alleen over het product maar ook over het proces. Aan het eind van het onderzoek zal er ook een evaluatiefase plaats vinden.

Intern -en multidisciplinair samenwerken:

Het project zal uitgevoerd worden volgens de Scrum methode. Voor deze methode zal er goed samengewerkt moeten worden.

Initiëren, organiseren en regisseren:

Voor deze competentie is Scrum ook van groot belang. Alle planningen zullen gedaan worden volgens een scrum planning.

Werkwijze

Voor dit project gaan we gebruik maken van de agile development methode Scrum. Dit project op de Scrum methode, het is daarom belangrijk dat alle deelnemers en partijen weten wat Scrum is. Zie voor een goede uitleg over de Scrum methode bijlage 1. Omdat we een aantal methodes gebruiken die nog niet bij alle deelnemers bekend zullen zijn, wordt er aan het begin van het project de tijd genomen om iedereen uit te leggen hoe alles werkt.

Scrum afspraken

- Digitaal Scrumboard en backlog via JIRA
- Elke sprint duurt 7 dagen waarbij het weekend dus ook mee telt als werkdagen. Aan het eind van elke sprint zijn alle taken afgerond en is er een werkende en geteste versie van de game. Bij de onderzoeksfase zullen er meerdere eindproducten per sprint zijn.
- Indicaties worden gegeven in storypoints volgens een Fibonacci schaal (1,2,3,5,8,13,20,40,60,100)
- Elke dag is er in de ochtend een standup volgens afspraak. Bij voorkeur is dit op een fysiek plek zoals school. Wanneer dit niet mogelijk is zal de standup digitaal plaats vinden via Google Hangout.
- Er wordt gebruik gemaakt van sprint burn down charts om te kijken of de sprints nog volgens planning gaan. Bij het ontwikkelen van de daadwerkelijke game wordt er gebruik gemaakt van een release burn down chart
- Er worden user stories gemaakt
- Sprint review en retrospective aan het eind van de sprint, waarbij alle teamleden presenteren wat ze gedaan hebben en het team gezamenlijk terug kijkt op de afgelopen sprint.

Github

Github wordt gebruikt voor de versiebeheer voor de game en prototypes. Dit systeem wordt gebruikt om overzichtelijk samen te kunnen werken. Er wordt altijd vanuit gegaan dat er een werkende versie op Github staat. Wanneer een projectlid taken heeft afgerond kan hij/zij zijn wijzigingen doorsuuren naar de centrale versie op Github. Het is daarom belangrijk dat teamleden goed communiceren waar ze mee bezig zijn zodat bestanden niet corrupt raken. Hierdoor worden alle projectleden aan het begin van het project op de hoogte gesteld hoe ze moeten werken met Github. Omdat de game open source wordt, is de Github, en dus de broncode, ook voor iedereen inzichtelijk.

Google Docs/Drive/Hangout

Documenten zoals verslagen en rapporten worden gemaakt in Google Docs. Google Docs biedt als voordeel dat er realtime met meerdere personen aan één document gewerkt kan worden waarbij iedereen altijd gesynchroniseerd is. Google Docs bestanden worden tevens gelijk gesynchroniseerd met Google Drive. Google Hangout wordt gebruikt voor (video)besprekingen zoals Scrum stand-ups of vergaderingen waarbij direct bestanden uit Google Drive en Docs getoond en bewerkt kunnen worden.

Alle bestanden die directe betrekking hebben tot de game hebben zoals code en game assets worden allemaal in Github opgeslagen. Alleen bestanden die geen directe betrekking tot de game hebben worden in Google Drive opgeslagen zoals een schets of een verslag.

Benodigdheden

- Google+ Account met toegang tot Google Drive, Google Docs, Google Hangouts en Google Communities.
- iOS developer account
- Xcode voor iOS development (+ mac osx computer)
- Android Development Account (alleen pas bij het publiceren in de Play Store)
- Testflight voor gebruikerstesten en onderzoek
- Development devices (minimaal 1 iOS en 1 Android device)
- JIRA (met account voor elk projectlid)
- Github repository (met account voor elk projectlid)
- Adobe Photoshop CS6 (Elk projectlid)
- Adobe Illustrator CS6 (Elk projectlid)

Communicatie

Intern

WhatsApp

Om snel intern met elkaar te communiceren wordt WhatsApp gebruikt. Er wordt een groep aangemaakt waar alle projectleden in komen. Dit is een informele manier om met elkaar te communiceren. WhatsApp wordt dan ook voornamelijk gebruikt om bijvoorbeeld aan elkaar te vragen waar iedereen is of om even iets kleins te melden.

Google+ Community

Een Google+ Community is het best te vergelijken met een Facebook groep. Op de Google+ Community worden bijvoorbeeld nuttige en/of inspirerende dingen gedeeld zoals video's, blogs, websites, tutorials en dergelijke. Daarnaast worden er ook deadlines op gezet ter herinnering.

Google Hangout

Google Hangout is een platform om vergaderingen en besprekingen te houden indien projectleden niet op een fysieke locatie bij elkaar (kunnen) zijn. In Hangout zijn een aantal dingen mogelijk zoals videovergadering (webcam), realtime gebruik maken van Google Drive en Google Docs inclusief real-time collaboration. Ook kan er gebruik worden gemaakt van Screenshare waardoor er één scherm gedeeld kan worden voor bijvoorbeeld uitleg van software en dergelijke. Hangout zal voornamelijk worden gebruikt voor als de Scrum stand-ups niet op een fysieke locatie gebeuren.

JIRA

Wordt gebruikt om de status van het Scrumproces te delen en bij te houden. Hier worden de taken en de status hiervan bijgehouden in een backlog en digitaal Scrumboard. Jira is vooral bedoelt als managementtool en niet als communicatie middel, het Scrumproces moet nog steeds besproken worden via standups.

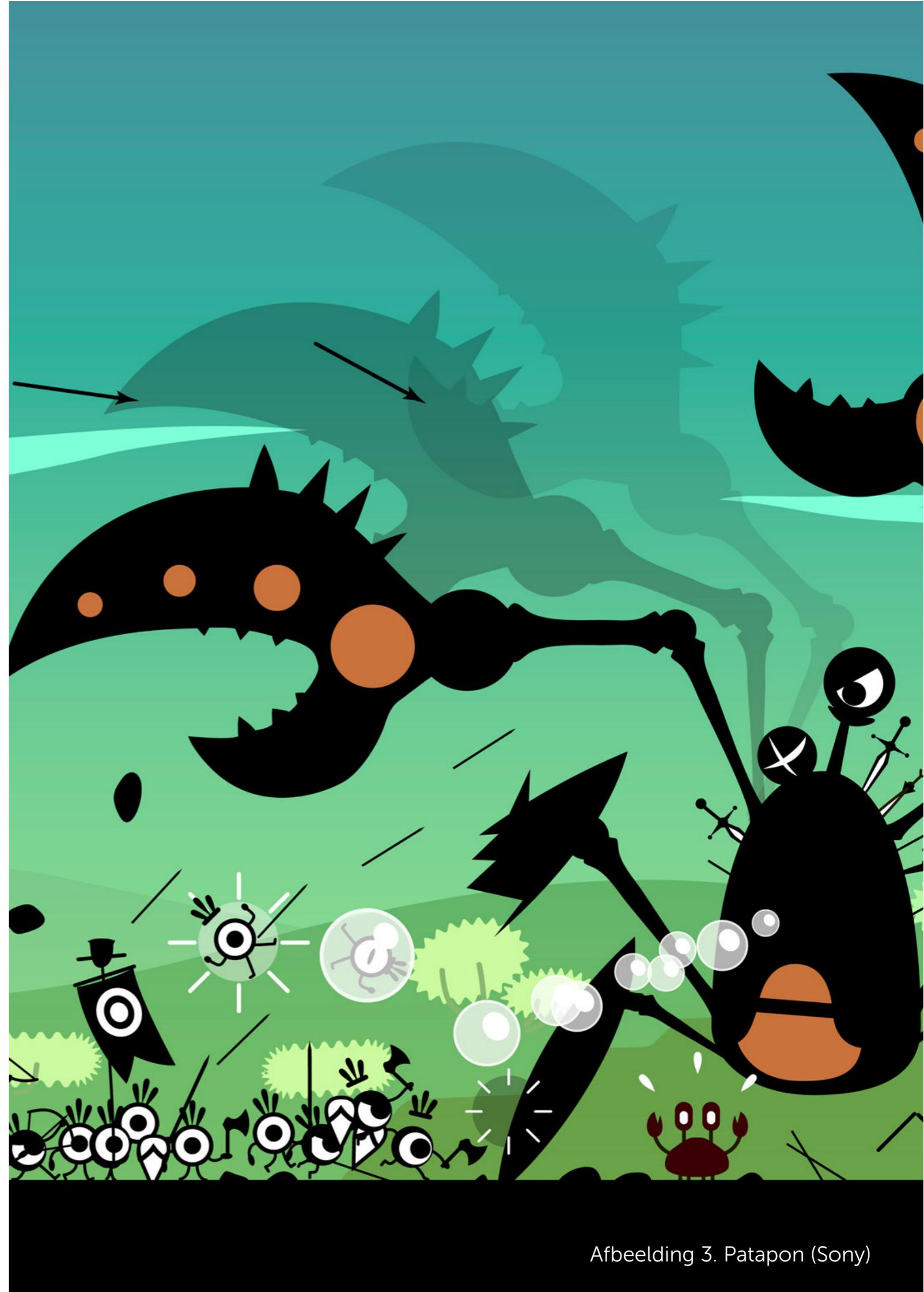
Extern

Avans email

Het Avans emailadres wordt gebruikt om emails te versturen naar derden. Dit kunnen project- en procesbegeleiders zijn, de verdiepingscoördinator maar ook experts of andere buitenstaanders waarmee contact moet worden opgenomen.

Blog

Het projectproces wordt voor buitenstaanders in kaart gebracht door middel van een blog. Voor de blog maken we gebruik van Tumblr. Het projectproces blog is niet gelijk aan de tool om je competenties te bewijzen voor de studieloopbaancoaching. Hiervoor werd ook een blog aangeraden. Om de competenties te bewijzen dient een projectlid een andere blog aan te maken.



Afbeelding 3. Patapon (Sony)

Kosten

De kosten worden afgehandeld door de projectleiders. Ze zullen ervoor zorgen dat de kosten verrekend zullen worden onder de groep. Het is daarom de individuele taak van een projectlid om gemaakte kosten zoals printwerk door te geven. Er wordt voor de ontwikkeling geen developers accounts aangeschaft. Alleen JIRA en bijbehorende software zal geregeld worden door de opdrachtgever.

- JIRA (\$20,- per maand)
- iOS developer account (\$99,- per jaar)
- Android Development Account (\$25,- eenmalig)

Een account voor de iOS Development Program of een Android Development Account is niet nodig voor het ontwikkelen van een game. Dit aangezien een developer account alleen nodig is voor de publicatie van de game. Een projectlid is altijd vrij om voor zichzelf een account aan te schaffen.

Verwachtingen van projectleden

Van de projectleden wordt verwacht dat ze actief participeren aan het project. Dit zijn vanzelfsprekende dingen die we extra benadrukken. Zo wordt er van projectleden geacht dat ze aanwezig zijn bij besprekingen. Projectleden moeten tegen werkdruk kunnen en zich houden aan deadlines. Werk dat opgeleverd wordt moet kwalitatief in orde zijn. Daarnaast moet een projectlid ook individueel kunnen werken.

De projectleiders hebben het recht om het projectlidmaatschap te onteigenen van een projectlid indien;

- er aanwijzingen zijn dat het projectlid meelift op het project;
- het projectlid geen kwalitatief werk oplevert;
- het projectlid zijn/haar deadlines niet haalt

Bij het niet goed functioneren van een projectlid kan deze een waarschuwing krijgen. Bij het krijgen van een tweede waarschuwing zal contact opgenomen worden met de projectbegeleider of stagecoördinator voor verdere stappen.

Derde partijen

Verdiepingscoördinator: Mikel Hanenkamp

Email: mrr.hanenkamp@avans.nl

- Toewijzen van een projectbegeleider met kennis over het onderwerp
- Tijdig goedkeuring geven voor het project
- Eventuele wijzigingen op tijd doorgeven

Projectbegeleider: nog niet bekend

Wekelijks gesprek over de status en voortgang van het project

Kennis over interactie, onderzoek en games

Rechten

De rechten van het product komen onder een nog nader te bepalen licentie. Het wordt naar alle waarschijnlijkheid een van de volgende licenties of een samenvoeging van licenties die compatibel met elkaar zijn (zo is bijvoorbeeld de EUPL compatibel met GNU (A)GPL). Hiervoor zal ook nog overleg plaats moeten vinden met Avans Hogeschool.

GNU AGPL (GNU Affero General Public License)

<http://www.gnu.org/licenses/agpl-3.0.html>

CC BY-SA 3.0 (Creative Commons Attribution-ShareAlike 3.0 Unported)

http://creativecommons.org/licenses/by-sa/3.0/deed.en_US

EUPL (European Union Public Licence)

http://en.wikipedia.org/wiki/European_Union_Public_Licence

Uiteindelijk wordt de game onder een opensource licentie gemaakt. Dit houdt in dat iedereen vrij is om de sourcecode te bewerken en te verspreiden onder de voorwaarde dat de bewerkte versie de zelfde licentie krijgt als het origineel. Dit houdt in dat als er een game maakt wordt door derden met onze code dat die game ook weer opensource moet worden. Daarnaast moet er credits gegeven worden aan de makers van het origineel.

Projectleden en de opdrachtgever staan vrij om dit project te gebruiken in hun portfolio. Alleen wanneer het projectlid een niet substantiële bijdrage geleverd heeft kan hij alleen deze bijdrage gebruiken voor zijn portfolio.

Contact

Jesse van Rheenen
0615215289
j.vanrheenen@student.avans.nl

Davey van der Woert
0653116446
dp.vanderwoert@student.avans.nl

Bijlagen

[Advertise Here](#)**(mt) mediatemple**(dv) DEDICATED-VIRTUAL
PRO-LEVEL VPS HOSTING
YOU'LL NEVER OUTGROW**GET STARTED**

plus

SCRUM: The Story of an Agile Team

Patkos Csaba on Jan 10th 2013 with [33 Comments](#) and [352 Reactions](#)

Tutorial Details

-
- **Difficulty:** Advanced
- **Completion Time:** 1 Hour

Scrum is one of the most heavily used agile techniques. It's not about coding; instead, it focuses on organization and project management. If you have a few moments, let me tell you about the team I work with, and how we adopted Scrum techniques.

A Little History

Scrum's roots actually extend beyond the Agile era.

Scrum's roots actually extend beyond the Agile era. The first mention of this technique can be found in 1986, by Hirotaka Takeuchi and Ikujiro Nonaka, for commercial product development. The first official paper defining Scrum, written by Jeff Sutherland and Ken Schwaber, was presented in 1995.

Scrum's popularity grew shortly after the 2001 publication of the [Agile Manifesto](#), as well as the book [Agile Software Development with Scrum](#), coauthored by Ken Schwaber and Mike Beedle.

A Few Facts

Scrum defines a set of recommendations, which teams are encouraged to follow. It also defines several actors – or roles, if you prefer that terminology – together with an iterative process of production and periodical planning. There are several tools, which accommodate the Scrum process. I will reference a few in this article, but the most powerful tools are the white board and sticky notes.

There is not, and never will be, a list of "Scrum Best Practices," because team and project context trumps all other considerations. — Mike Cohn

The Roles

Everything starts with the pig and the chicken. The chicken asks the pig if he is interested in jointly opening a restaurant. The chicken says they could call it, “Ham-and-Eggs.” The pig answers, “*No thanks. I’d be committed, but you’d only be involved!*”

That’s Scrum! It specifies a concrete set of roles, which are divided into two groups:

- **Committed** – those directly responsible for production and delivery of the final product. These roles include the team as a whole, its members, the scrum master, and the product owner.
 - **Involved** – represents the other people interested in the project, but who aren’t taking an active or direct part in the production and delivery processes. These roles are typically stakeholders and managers.
-

This is How We Started

Everything depends on dedication and good will. If you want your team to be efficient, productive, and deliver on time, you need someone to embrace some form of Agile techniques. Scrum may or may not be ideal for you, but it is surely one of the best places to start. Find that someone on your team who is willing to help the others, or you, yourself, can take on the responsibility of introducing Scrum.

You may ask why you should care how another team, like mine, does Scrum. You should care because we all learn how to do Scrum better by hearing stories of how it has been done by others – especially those who are doing it well. – Mike Cohn

The talented team I work with already knew a lot about Agile. We switched from Waterfall development to a more agile process, and released quite frequently. We successfully managed to release every three to six months, having a decently low number of bugs after each release.

But, still, we were far from what we can achieve today. We missed the process, or rules, that would force us to change our perspective on the product and process. That was the moment when our team manager introduced us to Scrum, a term we, at that time, had never heard of.

This person took the role of the **Scrum Master**.

The Scrum Master

The *Scrum Master* is easily one of the most important roles. This person is responsible for creating a bridge between the **Product Owner** (defined below) and the **Team** (also defined later). This person usually possesses a strong technical knowledge, and actively participates in the development process. He or she also communicates with the *Product Owner* about the **User Stories**, and how to organize the **Backlog**.

The Scrum Master coordinates development processes, but he does not micro-manage (the team is self-organized). At the beginning of the process, however, the Scrum Master might micro-manage part of the team, in order to improve their team interaction and self-organization techniques.

Scrum Masters have more responsibilities, and I’ll cover them as we discuss this process.

Introducing the Term “Sprint”

Personally, we don’t have any problem with three to six month releases, but I originally couldn’t imagine such a frequent release schedule. I thought it was too fast, and didn’t provide us with the necessary time to integrate and debug our products. But then, our Scrum Master introduced us to the term, **sprint**.

Sprint: a basic unit of development in Scrum. It can take between one week and one month, and the product is in a stable state after each sprint.

That sounds outrageous! Stable every week? Impossible! But, in actuality, it's quite possible. First, we reduced our production cycles from three months to one-and-a-half months, and, finally, to a single month. All of this was accomplished without changing our style. However, you'll need to introduce some rules for a sprint less than thirty days. There's no magic rule-set here; the rules must benefit your project.

If I recall correctly, the first significant change in our development process came by the introduction of **sprint planning**.

Sprint Planning

This is one of the several meetings that Scrum recommends. Before each new sprint, the team, product owner, and scrum master plan the next sprint. This meeting can take a whole day, but shorter sprints likely only need a couple hours or so.

Our process is mostly reviewing the product backlog, and deciding upon a subset of user stories that will be included in the next sprint. These decisions are made by direct negotiations between the team, represented by the scrum master, and the product owner.

The Product Owner

Setting the direction of a product by guessing which small feature will provide the most value may be the most difficult task.

This person is responsible for defining the *User Stories* and maintaining the *Product Backlog*. He or she is also a bridge between the team and higher management. The product owner evaluates the requests from stakeholders, higher management, users, and other feedback (like bug reports, user surveys, etc).

He or she *prioritizes* this backlog, providing the maximum value to the stakeholders in the shortest possible time. The product owner achieves this by planning the most valuable user stories that can be completed in a timely manner. This may sound sophisticated – it is! In fact, setting the direction of a product by guessing which small feature will provide the most value may be the most difficult task of the whole process. On the other hand, sometimes it's rather easy. You may have a thousand users asking for a specific feature. In these cases, the correct choice is obvious.

If those users represent a large portion of your user base, providing that specific feature increases loyalty.

But again, this is a difficult choice. What if you could increase your user base by 100% by implementing a different feature? So, you can either increase your current users' loyalty, or increase the user base. What is the correct choice? It all depends on the current direction of the business, and the product owner must decide where to take the product.

In the company I work for, these decisions propagate to the team. It's not a requirement of the Scrum process, but it is especially useful with new teams. Sharing information goes a long way in helping everyone understand why some decisions are made, and why seemingly obvious features may be delayed or dropped.

The Planning Board

I remember the morning it happened: I arrived at the office, only to find our scrum master preparing a makeshift planning board with A4 paper and transparent tape. I had no idea what he was doing. As with every morning, I

prepared a pot of coffee, and waited to see.

When finished, a white board was placed on the wall. It had several columns, and morphed into a rectangular shape. Several colored sticky notes peppered the “board.” That was two years ago.



The board now accommodates the **Lean Development Process** that we use today. Remember, Agile is all about change and adapting to change. Never blindly follow the rules.

So, what's on that board?

Columns for the Development Process

There are four main columns:

- **Release Backlog** – Where all the stories reside for the current release. Yes, the product is ready for release after each sprint, but that doesn't necessarily mean that we actually release it. We typically have five-day sprints.
- **Sprint Backlog** – When we plan, we negotiate what the product owner wants to complete in the sprint. How do we decide what we can and cannot complete? By estimating the difficulty of each story (see below – Estimating stories). The end result is a set of stories moved from the release backlog to the sprint backlog. The team concentrates on finishing those stories in the upcoming week.
- **Working On** – This one is simple. When team members take a story, they add it to this column, to show everyone what they are working on. This is meant for employee control, but rather for communicating with team members. Everyone should always know what their teammates are working on. In the above image, the small bookmarks stuck on the sticky notes contain my team members' names.
- **Done** – Complete all the things! Yes, this is where completed stories go. However, it's important to define “what is done.” At the end of an ideal sprint, all stories and bugs from the sprint backlog should be contained within this column.

Tip: Many teams like to split the **Working On** column into several others to better define different stages of work. We split ours into Design, Development and Testing. Feel free to invent your own steps, according to your process.

The Definition of Done

What is done? When can you confidently state that a story is complete? How do you know?

“Done” must be clearly and precisely defined, so that everyone knows when a story is complete. The definition of “done” can differ from team to team, and even from project to project. There is no exact rule. I recommend that you raise this issue at a team meeting, and decide what determines a complete story. Here are some ideas that you might want to consider:

- Create a minimalistic design.
- Create a GUI mockup.
- Use TDD or ensure that there are unit tests.
- Write the code.
- Let another team member manually test your story.
- The whole system can be built and compiled with the new code.
- Functional or acceptance tests pass as expected, after the new code is integrated into the system.

There are multiple ideas that can be included in the definition of done. Take what you consider to be necessary for your team and use it. Also, don’t forget to adapt this definition over time. If you notice that your definition is becoming outdated, you may consider removing some elements or adding necessary, but frequently forgotten, ideas.

In the picture above, the green sticky notes describe what we considered to be done, for each part of the process.

Populating the Board

That was the question we asked ourselves. Until this point, we did not use sticky notes for planning. We used software to keep track of user stories and bugs, but, other than that, we used nothing. After lunch, our scrum master presented us with a mountain of colored sticky notes. After preparing a dozen notes, he explained them to us.

The User Stories

A user story is a short, one sentence definition of a feature or functionality.

These represent the main features that we want to implement. A user story is a short, one sentence definition of a feature or functionality. It is referred to as a user story, because it is presented from the perspective of a user. Naturally, the term *user* is the person using our application. This person can be in one or more different categories: a system administrator, restricted user, manager, etc.

An example of such a story might sound something like, “*As a user, I want to share a folder with my teammates.*“

At that point, we did not have a product owner defined, so our scrum master invented these stories. This is acceptable at the beginning, but I **highly recommended** that you separate these two roles. Otherwise, how can the scrum master negotiate the sprint backlog with the product owner?

You may think to yourself, “*Why negotiate? Isn’t it actually better for a single person to decide what to do and when?*” Not quite. The two roles would be influenced by a single person’s views of the system or project. Two people, on the other hand, have a better chance of providing a more objective path for the team, ensuring the end goal (better valuable software) is achieved.

The product owner should define user stories; the team should negotiate their execution, and the scrum master represents them.

User stories define everything new that needs to be done; they're represented by yellow sticky notes on our board.

Bugs, Bugs, Bugs

Tracking your bugs is incredibly important.

We also list bugs on the board. Do you see those red sticky notes? Those are the bugs that we need to fix for the next release.

Different teams treat bugs in different ways. Our team mixes the bugs with the stories, but we always begin a sprint by fixing the bugs.

I know of other teams who pile up bugs for a period of three sprints, and spend every fourth sprint fixing them. Others split sprints into 25/75 for bugs/stories. Further, other teams may work on stories in the morning, and bugs after lunch; it simply depends on the company.

It's up to you to find the best solution for your team, and of course, keep track of your bugs. Write them on sticky notes so that you can track your system's issues and the fixes for those issues. Tracking your bugs is incredibly important.

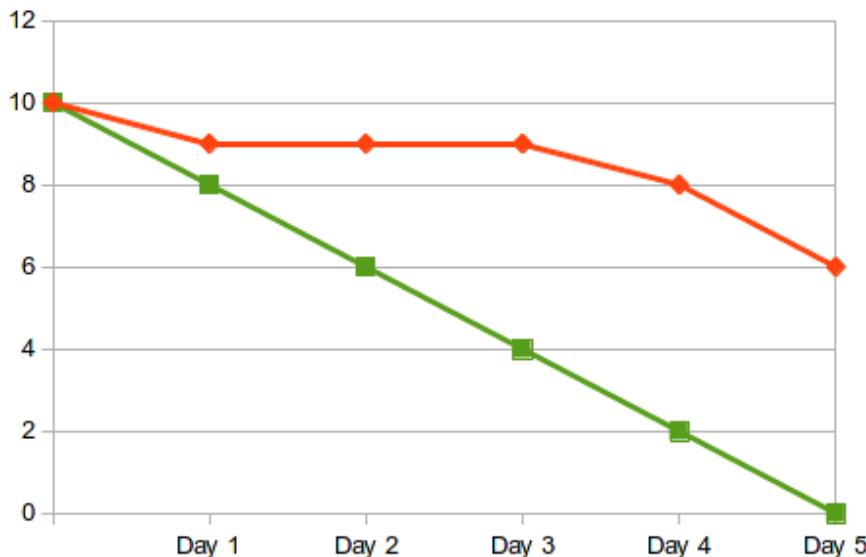
Tasks or Sub-Stories

Tasks are written as simple sentences from the developer's point of view.

Ideally, each story should be short enough to be completed with relative ease, but splitting stories into other stories can prove difficult. Some projects simply don't afford this possibility. Still, you'll find large stories that several team members can work on. It's important to divide huge chunks of work into smaller, easier to manage pieces.

One approach splits big stories into *Tasks*. Tasks are written as simple sentences from the developer's point of view. For example, the previous folder sharing story might be divided into several tasks, such as: "Create UI for sharing", "Implement public sharing mechanism", "Implement access control functionality", "Add Access Control checkboxes to the UI", and so on. The point is that you have to think more about the story every time you break it into smaller tasks. Your team will have much greater control over a story, when you analyze each piece.

We use light-blue sticky notes for tasks on our board. Look in the last column (the "Done" column), and you'll find our tasks under the user story stickies. That particular story was broken into around four tasks.



Technical Tasks

Certain activities must be completed in order to finish a task, story, or the sprint as a whole. These tasks are usually infrastructure related, but you may find that the tasks require changes to the system. This process may or may not be part of a story. For example, you may find that a third party application must be installed, in order to implement your application's sharing capability. Is this part of our user story? Possibly, but maybe not.

We determined that these tasks should be separated from the actual story. This helped us to better track these extra tasks. On our board, you can find these tasks on green sticky notes. There is one on the sprint backlog, and about three in testing.

The Technical Backlog

For a young team with little experience with Agile and Scrum, it's helpful to highlight these tasks with a mini-backlog.

This backlog is for infrastructural tasks, such as updating the automated testing system, configuring a new server, and other things, which make our everyday development work easier. These are things that must be completed at some point, but are not directly related to development.

You don't have to put these items into a separate backlog. In fact, I know of teams who don't separate them. Our team dropped our technical backlog a few months ago; we decided that infrastructural tasks are as important as other tasks. However, for a young team with little experience in Agile and Scrum, it's helpful to highlight these tasks with a mini-backlog. So, I recommend that you should give it a try. It may work for you, and, if not, then just put *Infrastructural Tasks* on your planning board, possibly with a different color.

The Big Challenge: Estimation

During a planning meeting, we decide which user stories and bugs from the product backlog (or release backlog in our case) to include in the next sprint. This may sound simple, but, in reality, it's quite complicated.

The product owner comes forward with a set of stories to work on. This list typically contains more work than what can be accomplished in the sprint. The scrum master, together with the team, has to convince the product owner of what can be done during a sprint. Over time, this process becomes easier, as the product owner learns the approximate velocity of the team. Then again, the team may become more productive with each sprint, thus allowing more stories to be finished. The trick is to have a team who really wants to exceed expectations!

Now, the product owner wants to complete more stories than we can do in a sprint. We need to estimate the amount of work we can do in relation to the stories submitted by the product owner, and we can do this in a variety of ways.

Story Points

Story Points are one of the most common methods for estimating stories, bugs or tasks. They are not necessarily the best approach, but they still are a good way to start.

So what is a story point? At the beginning of the process, the team looks for the simplest story they can find on the board. It doesn't matter how difficult it is, or how long it takes to complete. When they find that story, they set it as the reference story of having one point. In some projects, such a story can be as simple as fixing UI elements in ten minutes; whereas, the simplest story for more complex systems may take two hours for three people to complete. Now that you have a baseline, evaluate the other stories and bugs and assign them points.

This can be more difficult than it seems, and there are several point techniques to better help estimate stories. Here are two of them:

- Use Fobinacci Numbers – 1,2,3,5,8 (and maybe 13, but a 13-sized story smells too big to me).
- Use Powers of 2 – 1,2,4,8 (and maybe 16, but this number should be avoided).

You are free to choose whatever you feel most comfortable with. Be agile! Maybe you want to use two points increments, because your tasks are best estimated that way. Bravo to you!

Semaphores

Numbers are great, and many technical people love them. You may find, however, that, at some point, programmers begin to associate story points with time. They will think, “It takes me two days to do this. Let’s give it five points”. **This is wrong**. Estimates go from bad to worse when this happens. Story points should never relate to time.

Using semaphore colors may help alleviate this problem. Instead of assigning numbers to stories, the team can associate those stories with colors. Our team made this change a few months ago, and it greatly helped change our point of view.

Naturally, each color has a different meaning.

- Green signifies an easy task that can be completed in the next sprint.
- Yellow refers to a more difficult task – one that requires more effort from several team members. It also has a high chance of completion in the next sprint.
- Red labels are assigned to stories, which are extremely difficult and may not be finished in a single sprint. There should be little to no such stories, but if you adopt one week sprints, five days is a short time.

T-Shirt Sizes

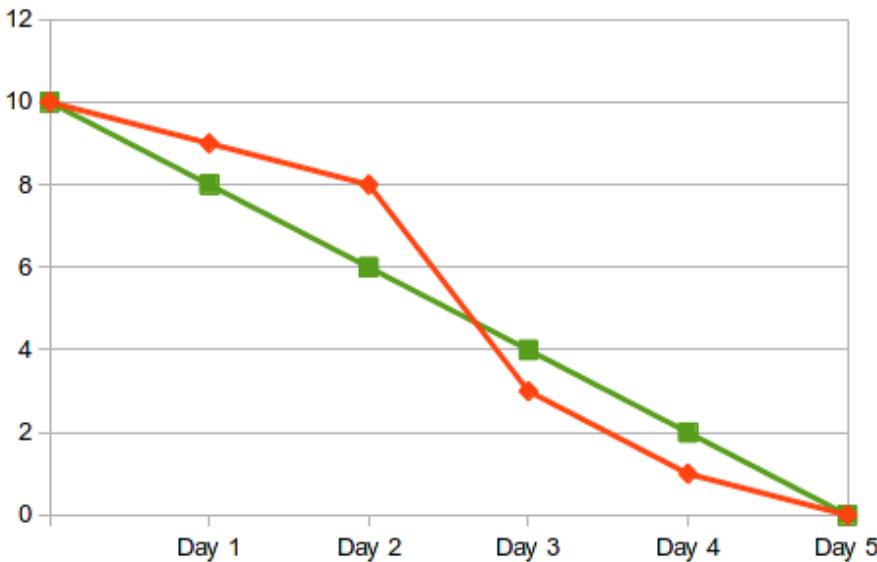
Numbers may be ugly to you, colors too artistic. It’s time for t-shirt sizes! This technique suggests giving up on comparing story complexity with time of completion. Simply put, instead of numbers, you use sizes like S, M, L, XL, XXL for your stories.

I personally never felt attracted to this kind of estimation, but, hey, some feel that it’s the best way to go. Try it out, if you feel comfortable with the idea.

The product owner, stake holders, and managers have to know what to expect from the end of a sprint. They must decide if the stories that were worked on should be released, and they have to know when features are ready. It’s not a good solution to release every new feature at the end of a product’s development cycle; releasing the most valuable features on a more frequent basis is a considerably better way to go. To achieve this, they must know what will be available in the short term, and their information should stem from the team. Therefore, the team should estimate, as best as possible, the work they can do in a sprint.

Measuring Speed of Development

So you want to see how well you perform in the current sprint? A frequently-used method is the burndown chart:

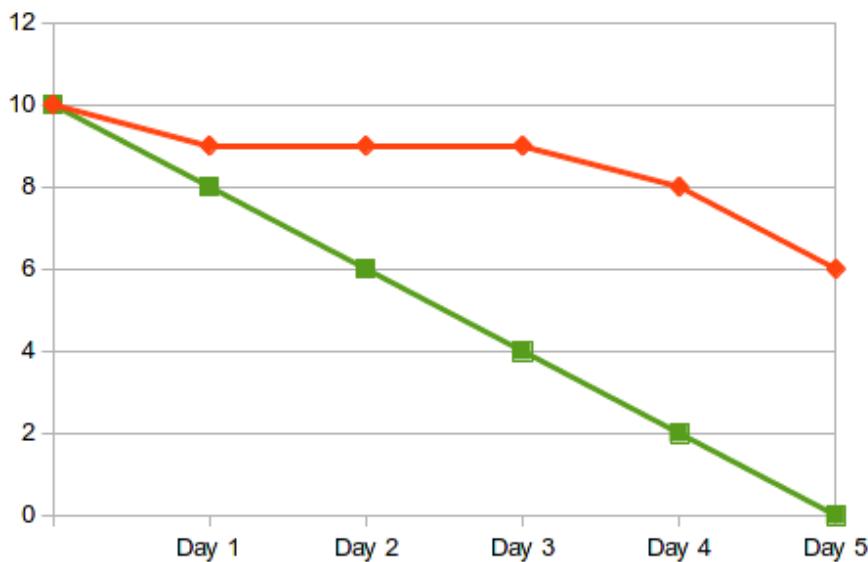


In this chart, we have a five day sprint, and we assumed that we could complete ten points in the sprint. Each value point represents the remaining story points at the end of each day. The green line reveals the ideal path: a steady two points per day. The red line shows our actual performance, or the true speed of development.

It's not on the planning board picture, but my team used to have an A4 paper sheet positioned above the planning board, with the burndown chart for each sprint. At the conclusion of each day, one of the team members was responsible for calculating the points completed for that day. It's simple: if programmers move the stories from column to column as they work, finding the remaining unfinished stories is easy.

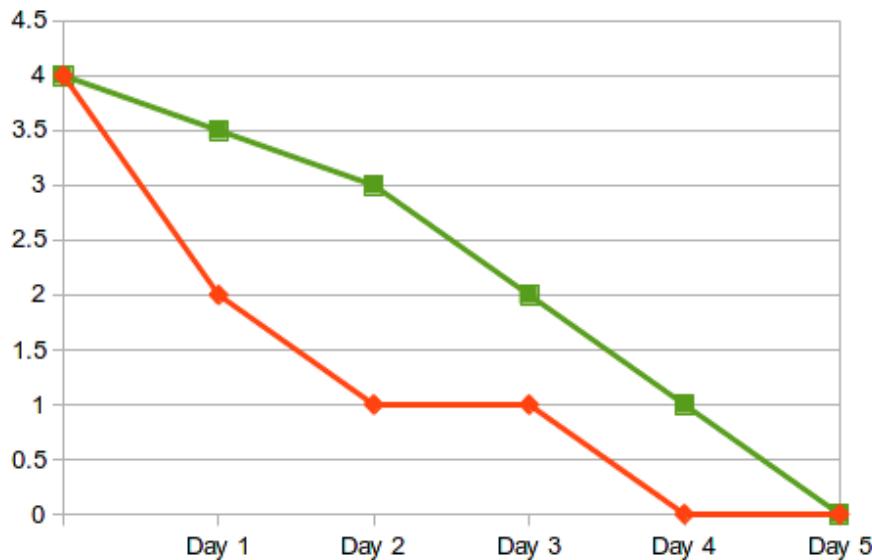
There are no half-done stories. If a story is done, it is done. If it is not complete, then it is not counted on the burndown chart.

Of course, you will fail – BIG TIME – at estimating! This is especially true at the beginning. There is no way to avoid this, unfortunately. There is no way to know how many points you can deliver. Your first chart may very well look like:



Our first chart surely looked similar to that. I think we did not even complete half of what we committed to. Why? Well, because estimation is hard. No matter what you do, or how good you are, when someone asks you how complicated something you have never done before is, you will have a hard time providing an accurate estimation. Don't panic! Try your best. With time, these things become easier. You may become able to estimate with a 70% accuracy at some point for short sprints. If the sprints are longer, your accuracy will likely be less, because there will be more stories to estimate and more variables that can go wrong.

When this happens, you adjust. For the next sprint, take four points. Like this:



This is bad again. You were too conservative, and finished early. It's a natural reaction for the team to adjust, based on the failure of the previous estimation. Still, this is a failure again, but on the other side of the road.

The problem? What do you do after you've finished what you were planning for? Another story? How do you put that on the chart? You can't redraw the original line.

When working with burndown charts, it's recommended to always make an average of the last 3-5 charts, in order to specify the points for the upcoming sprint. Of course, at the beginning, you don't have such information available, so you won't be as accurate as in the future. That's okay.

After some time, your charts will begin to resemble the first example more and more. You will, most of the time, finish all the stories and have a sustained velocity.

Velocity?

This term refers to your speed of development. It relates to how much you can do in a sprint. One of the most important concepts in Agile is to have a consistent velocity. Make the team deliver at a constant pace. With traditional project management, velocity decreases as a project ages. Complexity and rigidity forces the speed of development down.

Agile methodologies and techniques target to maintain a constant pace. Deliver fast now, and deliver faster later. How do they do it? Scrum is one of the elements of the puzzle. Other important pieces include the techniques that programmers can use to make their code and development process better. For example, XP (eXtreme Programming), Pair Programming, TDD (Test Driven Development), etc. All these, together, can make a team really great.

So we measure this velocity, but what do we actually do with it?

Tip: Measuring velocity is for making better predictions; not for judging a team or its members.

Use velocity to know what your team can do. Use velocity to know what to expect. Use velocity to make your team want more and be better. **Never** use velocity to judge your team or evaluate the productivity of your programmers.

Always Look Back and Improve

Following the first few sprints, our scrum master gathered the whole team. He began asking us about the good and bad things from the past week. This might be uncomfortable at the beginning, but it's still incredibly important. Describing what you felt went wrong in the past week will create awareness. And, of course, it is also helpful to highlight what went well!

These meetings are typically referred to as *Retrospectives*. They offer the scope to highlight what was good and what went wrong. Here are some examples from my own retrospectives.

Bad Things

- Team members were fighting too much
- Team member X or Y was not collaborative when pair programming
- We lost too much time with small things, like X or Y
- We did not pair program all the time
- We did not write unit tests for module M

When discussing problems, try to put aside your personal feelings; speak about what's bothering you. This is the only way for the team to resolve the issue. The most important thing is to immediately propose a solution for each problem. Don't simply let the list be written and forgotten; stay for a few minutes, and have the whole team think about what can be done to avoid them next week.

Good Things

- We finished on time
- We were able to talk without fighting
- Some of us became more receptive to suggestions and ideas
- We wrote all the code with TDD

Again, highlight as many good things as possible – especially at the beginning or with junior programmers. For example, having all the code written with TDD may be a big achievement for a junior team. Make them feel really good about it, make them want to do it more and more. The same is true for a senior team; they simply have other things to highlight (TDD is done by reflex).

I Want to See What You Did This Sprint

The demo is for showing stakeholders (and the product owner) the progress of the project.

This heading comes from the words of my scrum master. At that point, he also was the product owner. Before the end of a sprint, he'd ask us to present him with what we accomplished. We prepared a *Demo*, or a working example in a controlled environment.

Scrum proposes these demos at the end of each sprint. These should be done before the retrospective meeting that we discussed above. The team should prepare a special environment, and ensure that the product is capable of showcasing the features done in this sprint. The demo is for showing stakeholders (and the product owner) the progress of the project.

You may ask yourself why I mentioned a controlled environment, when our product should be production-ready at the end of each sprint. Yes, the product should be as close to production-ready as possible, but that does not mean that the feature, itself, is ready. Often, there will be features which are too big to fit within a single sprint. The product will remain stable, but the feature will not quite be ready. When stakeholders see the demo, they want to review the feature and what it can do. In many cases, to showcase some functionalities for unfinished features, special environments must first be prepared.

Additionally, based on these demos, the product owner may determine that a bigger feature is good enough, and a

new version of the product should be published and sent to the users. Most of the time, even if a feature is not quite complete, a release will help the project gain valuable user feedback, and concentrate the completion of the feature in a way that will satisfy as many users as possible.

Well, this sounds quite simple. You're an agile team, you keep your tests always on green, and your product is in a stable state. How difficult can it be to prepare a quick demo? It's more difficult than you might think!

Our team needed, if I remember correctly, more than five attempts before we managed to correctly prepare the demo. Luckily for us, the stake holders were not included in these first failed demos!

Still, We Need More Guidance

In these meetings, each team member must answer three questions.

It was the moment when our scrum master proposed to make a meeting each day. Yes! Every day, every morning, at an exact hour!

I find this to be a very good thing for new teams – for people who are not yet comfortable with one another, or with the project. These daily meetings, called *Daily Scrum*, are kept with the team, at a specified time every day. This should be kept before any work is done for the respective day. In my team, we set the time to 10AM each morning, which was difficult to do. Nonetheless, it was the correct decision.

The daily scrum is a short and simple meeting (not more than fifteen minutes). The scope of it is to help team members see who is doing what, and determine where the problems and bottlenecks in the development project are.

Tip: Because we want to ensure that these meetings remain short, we stand up. People usually get tired after 15 minutes of standing, which is perfect! If you find coworkers searching for places to sit and rest, your meeting has likely gone on too long.

In these meetings, each team member must answer three questions:

- **What did you do yesterday?** – A short answer is expected – max 2-3 sentences.
- **What are you planning to do today?** – Same type of short answer, things like “I will work on this story today.”
- **Are there any problems with your process?** If yes, what? Can they be quickly solved? This should be an answer highlighting the problems and solutions, if known. No detailed discussions should be taken, while this meeting goes on. The scrum master should take note of the problem, and work toward solving it together with the team, after the meeting is adjourned.

Solving the problems and impediments in the way of developers should be high priority for the team, so that they can continue with their development as soon as possible. Often, the person who had the problem is capable of solving it in a timely manner. Other times, he or she requires the help of a teammate. And other times, the problem can be so serious that the team will have to stop development and concentrate exclusively on solving the one thing that prevents them from continuing their work.

I remember my team encountering these huge road-blocks on several different occasions. There were tasks and stories, which seemed to be quite obvious at first site, but, after a pair or a single programmer had the chance to dig into the problem, the obvious became confusing and wrong. We discovered several times that a third party library could not provide us with the necessary functionality, and ended up concentrating all of our efforts into finding another, more capable library – or even implementing a solution ourselves.

The majority of our project is written in PHP. At some point, we had to interface our project with VMWare. We reviewed the official libraries for VMWare API, and found out there are Java and Perl versions. There's also an unofficial Ruby option. We were sure that we could use one of them, and simply do some `exec()` calls from PHP to capture the output as a string. As we thought, parsing from there should be piece of cake.

It turned out that this was next to impossible. Neither API library worked quite as we expected it to. Some were

abandoned or incomplete, and they had nearly impossible to parse outputs. Ultimately, we were forced to do something that nobody had ever done before: implement a VMWare API library in PHP. Unfortunately, there was no other reasonably acceptable way to do it.

This problem was massive; it set back the initial plans by weeks! Of course, our product owner was immediately notified, and, together with him, we planned a new schedule, and developed new stories, which included the creation of this API library.

More often than not, your problems will be much smaller. People might get stuck with some more sophisticated logic, but, many times, by the following morning, they already have ideas and solutions. Other times, they will simply be going on the wrong road, and a teammate will need to help get them back on track. These represent your typical issues.

Conclusion

Here we are at the conclusion. At least, this is how my team got started with Scrum. Some rules were very useful; others less so. Further, some rules were only useful for a short time, while others are still respected religiously by our team.

I can only recommend that you embrace the agile process, try out Scrum, and form your own conclusions. I'm sure that you'll find bits and pieces to adopt for your team. Be agile, adapt it for your style of work, for your projects and your personalities, and don't be afraid of adding your own custom rules. After all, Agile refers to adaptation, not blindly following a set of pre-determined rules!

For more top-shelf eBooks, courses, and tutorials, like this one, be sure to consider signing up for [Tuts+ Premium!](#)



353 people like this. Be the first of your friends.



Tags: [agile](#)[Premium](#)[scrum](#)

By Patkos Csaba

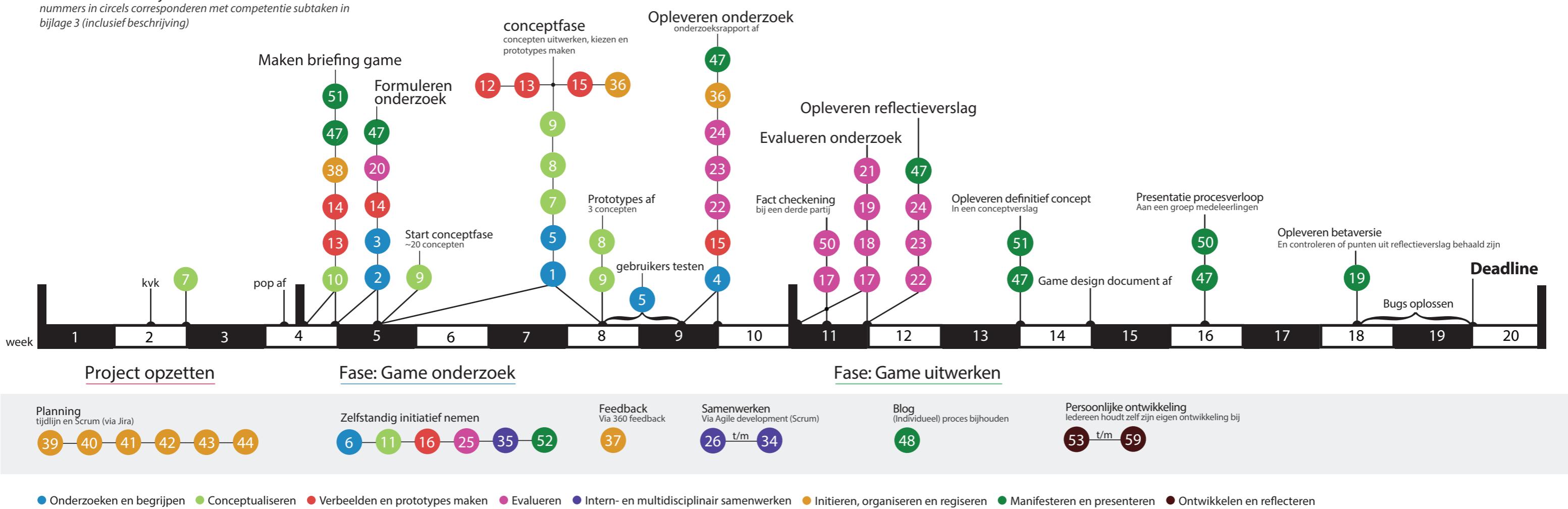
I am a passionate software developer with several years of experience especially with PHP, TDD and Agile Methodologies.

Note: Want to add some source code? Type `<pre><code>` before it and `</code></pre>` after it. [Find out more](#)

Bijlage 2: Tijdlijn blok 11 en 12

Mobile Game Project

nummers in cirkels corresponderen met competentie subtaken in bijlage 3 (inclusief beschrijving)



| A. ONDERZOEKEN EN BEGRIJPEN | | | |
|--|---|--|---|
| Kerntaak: Doen van onderzoek Toetsing: Beroepsproduct in de vorm van een onderzoeksverslag Beroepshandelingen: <ul style="list-style-type: none"> • Er is een debrief gemaakt; • Probleemstelling is SMART geformuleerd; • Er zijn minstens twee onderzoeksmethoden gebruikt om onderzoek te doen; • De gehanteerde onderzoeksmethoden zijn SMART omschreven; • Relevante onderzoeks vragen zijn SMART geformuleerd; • Er is geëxperimenteerd met minimaal twee middelen of methoden om antwoord te vinden op de geformuleerde probleemstelling; • Naar aanleiding van het onderzoek zijn SMART onderzoeks criteria geformuleerd; • De resultaten van het onderzoek zijn op een relevante manier geordend en sluiten aan bij de vraag/ probleemstelling; • De belangrijkste onderzoeksresultaten zijn herkenbaar in het eindresultaat. | | | |
| Voldoende | Goed | Hoe te bereiken | |
| De gebruikte onderzoeksmethoden, de vragen, de criteria of de ordening van resultaten sluiten aan bij de probleem/ vraagstelling. | De gebruikte onderzoeksmethoden, de vragen, de criteria of de ordening van resultaten zijn interessant/ van belang voor bepaalde personen, groepen of organisatie(s). | Door middel van een onderzoek naar (game)techniek, concept, doelgroep en interactie voor het spel, willen wij door middel van de resultaten het concept verbeteren in de concept/prototype-fase. | 1 |
| De onderzoeks vragen zijn relevant/ sluiten aan bij de vraag/ probleemstelling en zijn SMART geformuleerd; | De onderzoeks vragen zijn zeer relevant/ beschrijven volledig wat achterhaald dient te worden in relatie tot de vraag/ probleemstelling en zijn SMART geformuleerd; | In de briefing wordt in de briefings-fase gemaakt waarbij de onderzoeks vragen gedetailleerd in SMART geformuleerd worden met betrekking tot de vraag/ probleemstelling aan de hand van de opdracht criteria in de briefing. | 2 |
| De criteria sluiten aan bij de vraag/ probleemstelling en zijn SMART geformuleerd; | De criteria omvatten alle aspecten die gevraagd zijn in de vraag/ probleemstelling en zijn SMART geformuleerd; | De criteria die we zelf opstellen moeten geformuleerd worden voor ons onderzoek en voorkomen in het onderzoek en de onderzoeks vragen. | 3 |
| De resultaten van onderzoek zijn geordend. | De resultaten van onderzoek zijn geordend en helpen de lezer middels de gekozen structuur meer en snel inzicht te geven in de resultaten. | We ordenen de resultaten. Zorgen voor een overzichtelijk geheel. Bijvoorbeeld gebruik van infographics of andere visuele middelen. | 4 |
| Er heeft een iteratief onderzoeksproces plaatsgevonden. | Er is meermalen geïtereerd in het onderzoeksproces. | Na het vooronderzoek gaan we prototypes maken. Op basis van die prototypes gaan we verder onderzoeken en nieuwe prototypes maken (itereren). Op het einde van het project houden we een testronde met gebruikers. | 5 |
| Het onderzoek is deels onder begeleiding uitgevoerd. Eigen initiatief is getoond. | Het onderzoek is bijna helemaal zelfstandig uitgevoerd. Er was sprake van structureel eigen initiatief | Het is een eigen project met eigen initiatieven. Het onderzoek komt vanuit onszelf. We zetten die zelf op en gaan er zelf mee aan de slag. | 6 |

B. CONCEPTUALISEREN

Kerntaak: Ontwikkelen van ideeën tot een concept

Toetsing: Beroepsproduct in de vorm van een conceptverslag

Beroepshandelingen:

- Er zijn minstens twee methoden/ technieken ingezet om een idee te bedenken (experimenteren/divergeren);
- Verschillende ideeën zijn uitgewerkt (visueel, in schrift, auditief, tactiel, enz.) op basis van de opdrachtomschrijving;
- Er zijn onderbouwde keuzes gemaakt voor één of meerdere ideeën (convergeren);
- Ideeën zijn samengevoegd/ gecombineerd/ vergroot/ geordend tot een concept;
- Het concept sluit aan bij de criteria van de opdrachtomschrijving;
- Er zijn (mogelijke) verbeteringen op het concept geformuleerd.

| Voldoende | Goed | Hoe te bereiken | |
|---|---|---|----|
| Er is geëxperimenteerd met minstens twee tot drie methoden/ technieken en hiervan is materiaal; | Er is uitgebreid geëxperimenteerd met minstens vier methoden/technieken en hiervan is materiaal; | We gaan verschillende concepten bedenken en analyseren. We brainstormen voor concepten voor bijvoorbeeld het bedrijf en het spel. We maken mindmaps. X vs Y vs Z methode, ... | 7 |
| Er zijn minstens twee alternatieve concepten ontwikkeld; | Er zijn drie of meer alternatieve concepten ontwikkeld; | We maken veel verschillende concepten om vervolgens eliminatierondes te houden. Deze gebeuren door middel van keuzemethodes. | 8 |
| De keuzes die gemaakt zijn in het creatieve proces zijn onderbouwd met behulp van criteria uit de opdrachtomschrijving; | De keuzes die gemaakt zijn in het creatieve proces zijn onderbouwd met behulp van criteria uit de opdrachtomschrijving én eigen criteria; | Wij maken onze eigen briefing met daarin onze criteria. Onze keuzes baseren we op de criteria die in het begin zijn gesteld. | 9 |
| Het concept sluit aan bij de probleem/vraagstelling; | Het concept is interessant/van belang voor bepaalde personen, groepen of organisatie(s); | Het concept sluit zich aan op onze eigen briefing. Dit wordt bepaald tijdens het maken van de briefing. | 10 |
| De ontwikkeling van het concept is deels onder begeleiding uitgevoerd. Eigen initiatief is getoond. | De ontwikkeling van het concept is bijna helemaal zelfstandig uitgevoerd. Er was sprake van structureel eigen initiatief. | Het is een eigen project met eigen initiatieven. We werken structureel en we houden onze documentatie bij door bijvoorbeeld JIRA en/of een tijdlijn. | 11 |

C. VERBEELDEN EN PROTOTYPES MAKEN

Kerntaak: Ontwikkelen van een prototype*

Toetsing: Beroepsproduct in de vorm van een prototype*

(*Definitie prototype: een vormgegeven idee/concept wat visueel, taalkundig, auditief, tactiel of interactief kan zijn.)

Beroepshandelingen:

- Er zijn minstens twee mogelijkheden doordacht waarop een idee/concept in vormen en functies ‘vertaalt’ kan worden met behulp van multimediale technieken/ materialen/ gereedschappen ;
- Er is één prototype ontwikkeld met behulp van multimediale technieken/ materialen/ gereedschappen waarmee een idee/ concept vormen en functies gegeven worden;
- De keuze voor de multimediale vorm of functie kan onderbouwd worden vanuit criteria uit het idee/concept en de opdrachtomschrijving;
- Er zijn criteria opgesteld waarmee de functionaliteit/ verbeeldingskracht/ zeggingskracht van het prototype getest kan worden;
- Het prototype sluit aan bij de criteria van de opdrachtomschrijving;
- Er zijn (mogelijke) verbeteringen op het prototype geformuleerd.

| Voldoende | Goed | Hoe te bereiken | |
|--|---|--|----|
| Het idee/ concept is ‘vertaald’ naar een ‘tastbaar’ en testbaar artefact; | Het idee/ concept is ‘vertaald’ naar twee of meer ‘tastbare’ en testbare artefacten; | Wij gaan prototypes maken van verschillende concepten in verschillende vormen. | 12 |
| De keuze voor vorm en functie kunnen onderbouwd worden vanuit minstens drie tot vier criteria vanuit het concept of de opdrachtomschrijving; | De keuze voor vorm en functie kunnen onderbouwd worden vanuit minstens vijf criteria vanuit het concept of de opdrachtomschrijving; | In de briefing beschrijven we minstens 5 criteria voor vorm en functie. Hiermee houden we rekening bij het maken van de prototypes en het spel. | 13 |
| Er zijn criteria opgesteld waarmee het prototype getest kan worden; | Er zijn criteria opgesteld en methoden en technieken voorgesteld waarmee het prototype getest kan worden; | In de briefing worden eisen en criteria opgesteld om methode en technieken te bepalen voor het testen van de prototypes. In het onderzoek wordt onderzoek gedaan naar bijpassende en de beste methodes en technieken voor het testen. | 14 |
| Op basis van het testen van het prototype zijn minstens twee relevante verbeteringen in functionaliteit en vorm voorgesteld; | Op basis van het testen van het prototype zijn minstens drie relevante verbeteringen in functionaliteit en vorm voorgesteld; | Na het maken van prototypes itereren we deze. Daar bekijken we wat er beter kan in bijvoorbeeld vorm, functie en techniek. Na gebruikerstesten nemen we ook voorstellen voor verbeteringen van het concept. | 15 |
| De ontwikkeling van het prototype is deels onder begeleiding uitgevoerd. Eigen initiatief is getoond. | De ontwikkeling van het prototype is bijna helemaal zelfstandig uitgevoerd. Er was sprake van structureel eigen initiatief. | Het is een eigen project met eigen initiatieven. Hierbij ondernemen we zelf het initiatief voor het ontwikkelen van de prototype(s). | 16 |

D. EVALUEREN

Kerntaak: Opzetten, uitvoeren en analyseren van een evaluatie van een proces of product.

Toetsing: Beroepsproduct in de vorm van een evaluatierapport

Beroepshandelingen:

- Uit een opdrachtomschrijving worden de proces of productdoelen en de elementen die daarbij een rol spelen (zoals deelnemers, kwaliteit, tevredenheid, kennis, vaardigheid, vorming, affectiviteit, gedrag etc) op een rij gezet;
- Er worden evaluatiecriteria geformuleerd;
- Er worden evaluatiemiddelen en methoden gekozen om een proces of product te evalueren;
- De uitkomsten (data) uit de evaluatie worden geanalyseerd en gestructureerd;
- Er worden conclusies getrokken uit de evaluatie;
- Er worden verbeterpunten opgesteld voor het proces of product.

| Voldoende | Goed | Hoe te bereiken | |
|--|---|--|----|
| De proces of productdoelen en elementen die daarbij een rol spelen zijn herkent en eenduidig geformuleerd; | De proces of productdoelen en elementen die daarbij een rol spelen zijn herkent en eenduidig geformuleerd en bij de opdrachtgever, doelgroep, een professional of betrouwbare bron; factchecking; | Het onderzoek wordt gereflecteerd (ook door een derde partij). Daarmee wordt vervolgens een plan voor de toekomst gemaakt. | 17 |
| Er zijn eenduidige criteria geformuleerd om het proces of product te evalueren vanuit de opdrachtomschrijving; | Er zijn eenduidige criteria geformuleerd om het proces of product te evalueren vanuit de opdrachtomschrijving. Daarbij zijn aanvullende criteria geformuleerd; | De opdrachtomschrijving wordt aangepast aan de hand van het onderzoek. Daarbij worden nieuwe criteria opgesteld voor het definitieve concept. | 18 |
| Oorzaken van slagen of mislukken kunnen door de criteria achterhaald worden. | Oorzaken van slagen of mislukken en mogelijke verbeteringen kunnen door de criteria achterhaald worden. | Tijdens het evalueren kijken we naar eventuele mislukkingen, verbeterpunten en goede punten. Aan de hand van de criteria beoordelen we hoe dat zo is gekomen. | 19 |
| Er is van minstens twee evaluatiemiddelen of methoden gebruik gemaakt (bijv. enquête, interview, meting, vergelijking etc.) om het product of proces te evalueren; | Er is van minstens drie evaluatiemiddelen of methoden gebruik gemaakt (bijv. enquête, interview, meting, vergelijking etc.) om het product of proces te evalueren; | Bij het formuleren van het onderzoek en de onderzoek eisen wordt er nagedacht over evaluatiemiddelen zoals gebruikerstesten, A/B-testen et cetera. | 20 |
| De uitkomsten (data) volgen logischerwijs/zijn te herleiden uit de uitgevoerde evaluatie; | De uitkomsten (data) volgen helemaal uit de uitgevoerde evaluatie; | Tijdens evaluaties moet er objectief gebleven worden. Uitkomsten, conclusies en reflecties moeten worden gedaan op basis van de evaluatiemethodes. Op deze manier zorgen we er ook voor dat de data bruikbaarder wordt in vergelijking tot persoonlijke meningen en argumenten. Daarnaast kunnen we de data goed gebruiken om het project verder mee op te bouwen. | 21 |
| Resultaten zijn | Resultaten zijn geordend | Door middel van infographics maken wordt de | 22 |

| | | | |
|---|--|---|----|
| geordend; | en zijn makkelijk leesbaar en verhelderd door grafieken/ tabellen/ schema's/ highlights etc.; | data helder en gestructureerd. Hierdoor wordt het een overzichtelijk geheel die ook gebruikt kunnen worden om te reflecteren en verder te ontwikkelen. | |
| De conclusie volgt uit de resultaten of de conclusie geeft houvast voor het formuleren van verbeterpunten voor de vooraf opgestelde criteria; | <p>De conclusie volgt direct uit de resultaten of uit de conclusie kunnen direct verbeterpunten worden afgeleid;</p> <p>*Vage zin. Raar samengesteld. Lijkt alsof er iets mist of verkeerd is samengevoegd.</p> <p>**EVEN NAVRAGEN**</p> | De conclusie wordt tijdens de evaluatie gebaseerd op de resultaten van de evaluatiemethodes. Op basis van de conclusies worden verbeterpunten bepaald voor het project. Deze punten worden vervolgens meegenomen in de product- en/of proceseisen. | 23 |
| Er zijn minstens twee verbeterpunten geformuleerd voor het proces of product op basis van twee verschillende criteria; | Er zijn minstens drie verbeterpunten geformuleerd voor het proces of product op basis van drie verschillende criteria; | <p>Tijdens het evaluatieproces worden minstens 3 verbeterpunten geformuleerd voor het product of proces op basis van verschillende criteria.</p> <p>In het onderzoeksrapport worden verbeterpunten aangemerkt op basis van de vooraf opgestelde criteria.</p> | 24 |
| De opzet, uitvoering en analyse van de evaluatie is deels onder begeleiding uitgevoerd. Eigen initiatief is getoond. | De opzet, uitvoering en analyse van de evaluatie is bijna helemaal zelfstandig uitgevoerd. Er was sprake van structureel eigen initiatief | Het is een eigen project met eigen initiatieven. Hierbij willen we de evaluatie zelfstandig uitvoeren. | 25 |

E. INTER- EN MULTIDISCIPLINAIR SAMENWERKEN

Kerntaak: Samenwerken aan verschillende doelen in teams die in samenstelling van rollen en expertise verschillen.

Toetsing: reflectieopdracht in de vorm van een procesverslag en & 360° feedback.

Beroepshandelingen:

- Stelt zich in minstens drie situaties collegiaal op;
- Staat open voor feedback;
- Kan omgaan met kritiek;
- Geeft feedback en complimenten;
- Neemt een inhoudelijke rol in, in het team en wordt door de teamgenoten ook aangesproken in die rol;
- Creëert verhoogde productiviteit, kwaliteit en of betrokkenheid door samenwerking (1+1=3);
- Kent en neemt gezamenlijke verantwoordelijkheden;
- Bevordert het groepsproces;
- Leert de eigen invloed op het groepsproces kennen en kan daar minstens twee voorbeelden van noemen;
- Ontdekt wat de randvoorwaarden zijn voor een effectief/productief team (met wie, waar, hoe, wanneer) en kan daar minstens vier voorbeelden van noemen.

| Voldoende | Goed | Hoe te bereiken | |
|---|---|--|----|
| Uit minstens vijf eigen (STARR) voorbeelden, bewijsmaterialen en voorbeelden van teamgenoten, project- en procesbegeleider én opdrachtgever uit de 360° feedback blijkt dat de student: | Uit meer dan vijf eigen (STARR) voorbeelden, bewijsmaterialen en voorbeelden van teamgenoten, project- en procesbegeleider én opdrachtgever uit de 360° feedback blijkt dat de student: | Situatie – Wat speelt er Taak – Wat waren je taken Activiteiten – Wat heb je concreet gezegd/gedaan Resultaat – Wat gebeurde er daarna Reflectie – Stel je zelf vragen voordat je conclusies trekt. | |
| Zich in minstens drie situaties collegiaal heeft opgesteld; | Zich in meer dan drie situaties collegiaal heeft opgesteld; | Dit is een persoonlijke competentie. Daar moet individueel zelf aan voldaan worden. Dit kan eventueel getoond worden door de <i>stand-ups</i> met de Scrum-methode. | 26 |
| Open staat voor feedback; | Open staat voor feedback en daar ook om vraagt; geformuleerd; | Dit is een persoonlijke competentie deels gebaseerd op je karakter. Daar moet individueel zelf aan voldaan worden. Dit kan eventueel getoond worden door de <i>stand-ups</i> met de Scrum-methode die we gebruiken. | 27 |
| Met enige regelmaat feedback en complimenten geeft; | Vaak feedback en complimenten geeft en anderen zich daardoor geholpen en gewaardeerd voelen; | Dit is een persoonlijke competentie deels gebaseerd op je karakter. Daar moet individueel zelf aan voldaan worden. Dit kan eventueel getoond worden door de <i>stand-ups</i> door middel van de Scrum-methode die we gebruiken. Door te vragen wat een ander van de feedback kan je er achter komen hoe nuttig je feedback was. | 28 |
| Een inhoudelijke rol inneemt in het team; | Een inhoudelijke rol inneemt in het team en door de teamgenoten daar ook actief op wordt aangesproken; | Dit is een persoonlijke competentie. Daar moet individueel zelf aan voldaan worden. Er worden aan de projectleden wel een taak meegegeven. Het is aan de projectleden individueel hoe ze dit invullen voor hun competenties. Met Scrum zijn er iedere ochtend <i>stand-ups</i> . Daarbij wordt actief met elkaar gesproken en besproken. | 29 |
| Door samenwerking | Door samenwerking | Door het samenwerken (door middel van Scrum) | 30 |

| | | | |
|---|---|--|----|
| verhoogde productiviteit, kwaliteit en of betrokkenheid creëert; | verhoogde productiviteit, kwaliteit en of betrokkenheid creëert; | met JIRA) wordt de productiviteit verhoogd. De kwaliteit kan verhogen doordat meerdere personen aan één onderdeel werken maar juist ook door een goede taakverdeling door middel van JIRA. We willen de samenwerkingen erg ondersteunen door gebruik te maken van JIRA. | |
| De gezamenlijke verantwoordelijkheden kent en neemt; | De gezamenlijke verantwoordelijkheden kent en neemt. Anderen hebben het gevoel dat ze op hem/haar kunnen bouwen; | Dit kan eventueel getoond worden door de <i>stand-ups</i> met de Scrum-methode. Door de <i>sprints</i> in Scrum af te gaan kan er aan projectgenoten aangetoond worden dat je in staat bent om je werk te doen. Daarnaast worden er in de <i>stand-ups</i> intensief met elkaar gesproken. Hiermee wordt vertrouwen opgebouwd. Met Scrum wordt er in overleg bepaald wie welke taak neemt en wat de verantwoordelijkheden zijn die je hebt voor anderen. | 31 |
| Het groepsproces bevordert; | Het groepsproces helpt sturen; | Dit kan eventueel getoond worden door de <i>stand-ups</i> door middel van de Scrum-methode. Met JIRA kun je taken verdelen, nieuwe taken aanmaken en opmerkingen plaatsen voor bijvoorbeeld dealines en andere belangrijke gevallen waarop gelet moet worden. Iedereen is vrij om het groepsproces te bevorderen en te sturen. JIRA is er een hulpmiddel voor maar het licht bij de persoon zelf om dit daadwerkelijk te doen. | 32 |
| De eigen invloed op het groepsproces heeft leren kennen en kan daar minstens twee voorbeelden van kan noemen; | De eigen invloed op het groepsproces heeft leren kennen en kan daar meer dan twee voorbeelden van kan noemen; | Dit kan eventueel getoond worden door de <i>stand-ups</i> met de Scrum-methode. Als je eigen initiatieven op dit gebied hebt genomen kun je deze terug zien in JIRA. | 33 |
| Heeft ontdekt wat de randvoorwaarden zijn voor een effectief/ productief team (met wie, waar, hoe, wanneer) en daar minstens vier voorbeelden van kan noemen. | Heeft ontdekt wat de randvoorwaarden zijn voor een effectief/ productief team (met wie, waar, hoe, wanneer) en daar meer dan vier voorbeelden van kan noemen. | Dit kan eventueel getoond worden met JIRA als hulpmiddel. Door naar de <i>sprints</i> te kijken kun je zien wat is gehaald en wat nog niet is gehaald in een <i>sprint</i> . Hiermee kun je in JIRA analyseren wat de randvoorwaarden zijn voor een productief en effectief projectteam. | 34 |
| Verder: Het afronden van het inhoudelijk en procesmatig samenwerkingsproces is deels onder begeleiding uitgevoerd. Eigen initiatief is getoond. | Verder: Het afronden van het inhoudelijk en procesmatig samenwerkingsproces is bijna helemaal zelfstandig uitgevoerd. Er was sprake van structureel eigen initiatief. | Het is een eigen project met eigen initiatieven. Hierbij willen we de samenwerking zelfstandig uitvoeren. | 35 |

F. INITIEREN, ORGANISEREN EN REGISSEEREN

Kerntaken: Managen van jezelf en/of anderen en een project binnen kaders (tijd, budget en kwaliteit).

Toetsing: Reflectieopdracht in de vorm van een procesverslag & 360° feedback.

Beroepshandelingen:

- Koppelt interpretatie van wensen en doelen van de opdrachtgever minstens één keer terug ter controle;
- Maakt afspraken over werk- en rolverdeling en komt deze na (SMART geformuleerd);
- Formuleert meetbare en haalbare doelen (SMART geformuleerd);
- Stelt een planning op en stelt deze bij (SMART geformuleerd);
- Maakt werkafspraken (SMART geformuleerd);
- Stelt prioriteiten (SMART geformuleerd);
- Maakt keuzes over tijd, budget en kwaliteit in relatie tot de doelstelling van het project);
- Stelt verwachtingen bij;
- Kan op basis van minstens drie ervaringen en drie bewijsstukken concluderen wat positieve bijdragen waren aan het verloop van het project (planning, uitvoering, controle, beheer en informatie). (beschrijving volgens de STARR methode);
- Kan op basis van minstens drie ervaringen en de 360° feedback aantonen en daarmee concluderen wanneer, met wie, hoe, waar en op welke manier hij zichzelf en eventueel ook, afhankelijk van de rol, ook anderen het meest effectief managed (leidt, organiseert, beïnvloedt, motiveert) (beschrijving volgens de STARR methode).

| Voldoende | Goed | Hoe te bereiken | |
|---|--|---|----|
| Heeft de interpretatie van wensen en doelen van de opdrachtgever teruggekoppeld; | Heeft de interpretatie van wensen en doelen van de opdrachtgever teruggekoppeld en daarna bijgesteld; | Na het ontwikkelen van prototypes worden verwachtingen, wensen en doelstellingen bijgesteld. Na de gebruikerstesten wordt er op basis van de resultaten van het onderzoek terugkoppelingen gegeven over de wensen, verwachtingen en doelen van de briefing. | 36 |
| Kan met behulp van 360° feedback aantonen dat hij zich aan afspraken heeft gehouden over werk- en rolverdeling; | Kan met behulp van 360° feedback aantonen dat hij zich aan afspraken heeft gehouden over werk- en rolverdeling en daarbij rekening heeft gehouden met werkzaamheden en rollen van anderen; | Door middel van 360° feedback van onder andere projectparticipanten. | 37 |
| Doelen zijn meetbaar of haalbaar geformuleerd; | Doelen zijn meetbaar en bleken achteraf ook haalbaar geformuleerd; | We moeten in de briefing het project SMART formuleren. | 38 |
| De planning is uitgewerkt en bijgesteld tijdens het proces; | Heeft een planning uitgewerkt en bijgesteld tijdens het proces en heeft anderen hierover op de hoogte gesteld; | Er is een globale planning opgesteld in het begin van het project. Deze bijgesteld en bijgehouden wanneer dat nodig is. Daarnaast gebruiken we een Scrum-methode, door middel van JIRA, om de planning op orde te houden en eventueel bij te stellen wanneer dit nodig is. Door middel van het Scrum-bord op JIRA kan iedereen elkaar op de hoogte stellen. | 39 |
| Er worden prioriteiten gesteld. Hoofd- en bijzaken worden gescheiden; | Er worden prioriteiten gesteld. Hoofden bijzaken worden gescheiden; | Door middel van Scrum in JIRA wordt er een prioriteitsindicatie weergeven. | 40 |
| Keuzes over tijd, budget | Keuzes over tijd, budget | Door agile-development (Scrum), kunnen we | 41 |

| | | | |
|--|---|--|----|
| en kwaliteit worden in relatie tot elkaar of tot de doelstelling van het project bepaald /gewogen; | en kwaliteit worden in relatie tot elkaar of tot de doelstelling van het project bepaald /gewogen. Hierover werd met de opdrachtgever gecommuniceerd; | flexibel zijn in tijd, kwaliteit en budget. Dit komt omdat wij werken met vaste <i>sprints</i> . Deze <i>sprint</i> leveren elke keer een voorbepaald eindproduct op. Daardoor kan over elke <i>sprint</i> bepaald worden waarop gefocust kan worden. Hiermee wegen dus tijd, budget en kwaliteit tot elkaar af. | |
| Bijgestelde verwachtingen worden aan anderen of opdrachtgever gecommuniceerd; | Er wordt structureel met de opdrachtgever overlegd om zo verwachtingen bij te stellen; | Tussen de <i>sprints</i> is er een review moment waarin we bepalen wat er moet gaan gebeuren en wat er gedaan is. Hiermee stellen we onze verwachtingen bij. | 42 |
| Kan op basis van minstens drie ervaringen en drie bewijsstukken aantonen wat zijn positieve bijdragen waren aan het verloop van het project (planning, uitvoering, controle, beheer en informatie); | Kan op basis van meer dan drie ervaringen en meer dan drie bewijsstukken aantonen wat zijn positieve bijdragen waren aan het verloop van het project (planning, uitvoering, controle, beheer en informatie); | Door middel van JIRA valt al veel te bewijzen wat positieve bijdragen waren aan het project. Daarnaast kan dit ook met prototypes die gemaakt zijn. | 43 |
| Kan op basis van minstens drie ervaringen en de 360° feedback aantonen en daarmee concluderen wanneer, met wie, hoe, waar en op welke manier hij zichzelf en eventueel ook, afhankelijk van de rol, anderen het meest effectief managed. | Kan op basis van meer dan drie ervaringen en de 360° feedback aantonen en daarmee concluderen wanneer, met wie, hoe, waar en op welke manier hij zichzelf en eventueel ook, afhankelijk van de rol, anderen het meest effectief managed. Hierbij wordt verwezen naar een managementstijl. | Dit onderdeel kan onder andere bewezen worden door JIRA en 360° feedbacks. | 44 |
| Het managen van zichzelf en/of anderen en een project binnen gestelde kaders (tijd, budget en kwaliteit) is deels onder begeleiding uitgevoerd. Eigen initiatief is getoond. | Het managen van zichzelf en/of anderen en een project binnen gestelde kaders (tijd, budget en kwaliteit) is bijna helemaal zelfstandig uitgevoerd. Er was sprake van structureel eigen initiatief. | Het is ons eigen project. Hierbij managen we ons en andere in het project binnen de kader door onder andere gebruik te maken van JIRA. | 45 |

G. MANIFESTEREN EN PRESENTEREN

Kerntaak: Jezelf (multimidiaal) profileren aan een opdrachtgever, docenten, doelgroep of teamgenoten waarmee je enthousiasmeert en overtuigt.

Toetsing: Gedragsproef* en 360° feedback

* Dit kan zijn: presentatie/ pitch/ voordracht/ bespreking/ bijeenkomst/ vergadering / evenement

Beroepshandelingen:

- Ontwikkelen van een (show)portfolio;
- Presenteren en pitchen;
- Verhelderen of overtuigen d.m.v. metaforen/ verwijzing naar ‘grote’ verhalen;
- Argumenteren;
- Gebruik van sociale media om vakmanschap en visie te communiceren;
- Netwerken;
Keuze maken voor (multimedia/ communicatie) middelen of technieken waarmee je een opdrachtgever, docenten, een doelgroep of teamgenoten overtuigt of enthousiasmeert voor je vakmanschap, een boodschap/standpunt, een product of eindresultaat;

| Voldoende | Goed | Hoe te bereiken | |
|--|---|---|----|
| Communicatievaardigheden worden voldoende beheerst (kunnen omgaan met spreekstress, bewust stemgebruik, hoofd- en bijzaken kunnen scheiden, contact maken met de (behoefte van) toehoorders of doelgroep, zich voldoende voorbereiden, nagaan of de boodschap is overgekomen, argumenten doen ter zake etc.) | Communicatievaardigheden worden goed beheerst (weinig tot geen last van spreekstress, effectief stemgebruik, hoofd- en bijzaken kunnen scheiden, antwoord geven op de (behoefte van) toehoorders of doelgroep, zich goed voorbereiden, nagaan of de boodschap is overgekomen, | Tussen de helft van eind van blok 14 (studiejaar 2012/2013) is het de bedoeling om een expertles te geven over het opzetten van een bedrijf en het zelfstandig aanpakken, organiseren en regisseren van een projecten. Veel mensen willen wel iets voor zichzelf beginnen, maar de stap naar het onbekende, dus uit de confortzone, is voor hen te groot. Met deze expertles willen we ze inspiratie geven door middel van onze aanpak. | 46 |
| In de communicatie wordt gebruik gemaakt van metaforen of ‘grote’ verhalen om de boodschap te verhelderen of te overtuigen en ze sluiten aan op de kern van de boodschap; | In de communicatie wordt doelmatig/gericht gebruik gemaakt van metaforen of ‘grote’ verhalen om de boodschap te verhelderen of te overtuigen en ze sluiten aan op de kern van de boodschap; | In rapporten, verslagen, evaluaties en andere documenten maar ook in de expertles presentatie moeten er heldere metaforen gebruikt worden. Hiermee zorgen we ervoor dat de boodschap goed en duidelijk overkomt. | 47 |
| Er wordt gebruik gemaakt van sociale media om vakmanschap en visie te communiceren; | Er wordt gebruik gemaakt van sociale media om vakmanschap en visie te communiceren en contacten te leggen; | Met een persoonlijke weblog kan visie en vakmanschap getoond worden door het tonen van gemaakte producten of doorlopen processen. Daarnaast is het aan een projectlid zelf om contacten te leggen en te netwerken via social media. Dit kan bijvoorbeeld met Linkedin, Facebook of Twitter. | 48 |
| Het showportfolio bestaat uit producten en processen waarmee onderscheidend, enthousiasmerend, kenmerkend, profilerend vakmanschap of visie op het beroepenveld getoond wordt. | Het showportfolio bestaat uit producten en processen waarmee onderscheidend, enthousiasmerend, kenmerkend, profilerend vakmanschap of visie op het beroepenveld getoond wordt; | Het persoonlijk portfolio wordt verder ontwikkeld door de opdracht. Daarbij is het aan een projectlid zelf om een visie te tonen, te profileren en te onderscheiden. | 49 |
| Er kunnen minstens twee | Er kunnen minstens twee | Door de factchecking voor het | 50 |

| | | | |
|---|---|---|----|
| bewijzen geleverd worden waaruit blijkt dat er genetwerkt is; | bewijzen geleverd worden waaruit blijkt dat er genetwerkt is; | evaluatieproces moet er genetwerkt worden. Daarnaast kan er zelf initiatief getoond worden tijdens het project op het gebied van netwerken waar het project baat bij heeft. Daarnaast moet er genetwerkt worden om ervoor te zorgen dat de explorties presentatie geven kan worden. | |
| De keuze voor (multimedia/communicatie) middelen of technieken om een opdrachtgever, docenten, een doelgroep of teamgenoten te overtuigen of enthousiasmeren is bewust gemaakt. Voor de keuze worden minstens drie argumenten geleverd die in lijn zijn met het communicatiedoel; | De keuze voor (multimedia/communicatie) middelen of technieken om een opdrachtgever, docenten, een doelgroep of teamgenoten te overtuigen of enthousiasmeren is bewust gemaakt. Voor de keuze worden minstens drie argumenten geleverd die in lijn zijn met het communicatiedoel; | Tijdens de onderzoek maar ook tijdens het proces moeten er bepaalde keuzes gemaakt worden. Het ligt aan een projectlid zelf om een ander te willen overtuigen indien hij/zij denkt dat een bepaalde methode de beste is. | 51 |
| Het profileren naar een opdrachtgever, docenten, doelgroep of teamgenoten om te enthousiasmeren en overtuigen is deels onder begeleiding uitgevoerd. Eigen initiatief is getoond. | Het profileren naar een opdrachtgever, docenten, doelgroep of teamgenoten om te enthousiasmeren en overtuigen is bijna helemaal zelfstandig uitgevoerd. Er was sprake van structureel eigen initiatief. | Het is een eigen project met eigen initiatieven. Hierbij moet het profileren zelfstandig gebeuren. | 52 |

| H. ONTWIKKELEN EN REFLECTEREN | | | |
|---|--|--|----|
| Kerntaak: Je blijven ontwikkelen als professional Toetsing: Reflectieopdracht in de vorm van een POP & 360° feedback. Beroepshandelingen: <ul style="list-style-type: none"> • Kunnen benoemen van kwaliteiten, valkuilen, uitdagingen en allergieën; • Laat door het formuleren van SMART leerdoelen zien dat hij/zij bewust onbekwaam is (weten wat je niet weet); • Doet aan intervisie met vak/studiegenoten; • Leert van feedback en kritiek; • Past reflectiemethoden toe om zichzelf bij te sturen op het gebied van kennis, vaardigheid en houding tijdens het proces van vraagstelling tot eindresultaat; • Leest vakliteratuur en volgt trends binnen het vakgebied; • Neemt deel aan sociale activiteiten en netwerken binnen het vakgebied; | | | |
| Voldoende | Goed | Hoe te bereiken | |
| Er worden minstens drie voorbeelden gegeven (mbv de STARR methode) van situaties of producten uit de verdieping waaruit de eigen kwaliteiten, valkuilen, uitdagingen en allergieën blijken; | Er worden meer dan drie voorbeelden gegeven (mbv de STARR methode) van situaties of producten uit de verdieping waaruit de eigen kwaliteiten, valkuilen, uitdagingen en allergieën blijken; | We zijn zelf verantwoordelijk voor zelfreflecties. | 53 |
| Er zijn minstens drie SMART doelen geformuleerd in het POP waaruit blijkt dat de student zich bewust is van eigen onbekwaamheden (weet wat hij/zij niet weet); | Er zijn meer dan drie SMART doelen geformuleerd in het POP waaruit blijkt dat de student zich bewust is van eigen onbekwaamheden (weet wat hij/zij niet weet); | We dienen zelf een POP op te stellen en daar zelfstandig aan te werken. | 54 |
| Heeft gedurende het project minstens één maal aan intervisie gedaan met vak/studiegenoten en heeft op basis daarvan de leerdoelen SMART aangepast; | Heeft gedurende het project meer dan één maal aan intervisie gedaan met vak/studiegenoten en heeft op basis daarvan de leerdoelen SMART aangepast; | Een projectlid is zelf verantwoordelijk om initiatief te ondernemen om een gesprek te organiseren indien daar wens naar is. Daarnaast zijn er <i>stand-ups</i> die bij de Scrum-methode horen. | 55 |
| Uit de 360° feedback blijkt dat de student leert van feedback en/of kritiek; | Uit de 360° feedback blijkt dat de student leert van feedback en/of kritiek en dat ook toepast; | We zijn zelf verantwoordelijk om feedback te vragen indien we niet genoeg feedback hebben gehad van projectleden, docenten of medestudenten. | 56 |
| Past minstens twee reflectiemethoden toe en gebruikt die om zichzelf bij te sturen op het gebied van kennis, vaardigheid en/of houding tijdens het proces van vraagstelling tot eindresultaat; | Past meer dan twee reflectiemethoden toe en gebruikt die om zichzelf bij te sturen op het gebied van kennis, vaardigheid en/of houding tijdens het proces van vraagstelling tot eindresultaat; | We zijn hier zelf individueel verantwoordelijk voor. Dit kan tevens gebruiken voor zijn/haar POP. | 57 |
| Leest vakliteratuur, bezoekt congressen/ evenementen/ tentoonstellingen/ bijeenkomsten waardoor trends binnen het vakgebied gevolgd worden en kan minstens | Leest vakliteratuur, bezoekt congressen/ evenementen/ tentoonstellingen/ bijeenkomsten waardoor trends binnen het vakgebied gevolgd worden en kan meer dan | Hier is iedereen zelf verantwoordelijk voor. Het is handig om dit bij te houden op een blog. Het kan zijn dat er gezamenlijk initiatief wordt genomen om naar een meeting te gaan van Adobe User Groups of naar bijvoorbeeld Google Analytics User Conference. | 58 |

| | | | |
|--|--|--|----|
| één voorbeeld geven van een idee, product of manier van werken waarin hij/zij zich heeft laten inspirerend door een trend of hedendaags inzicht; | één voorbeeld geven van een idee, product of manier van werken waarin hij/zij zich heeft laten inspirerend door een trend of hedendaags inzicht; | | |
| De ontwikkeling als professional heeft deels onder begeleiding plaatsgevonden. Eigen initiatief is getoond. | De ontwikkeling als professional is bijna helemaal zelfstandig uitgevoerd. Er was sprake van structureel eigen initiatief. | Het is een eigen project met eigen initiatieven. Het ligt bij de student zelf om te bepalen of ze hun persoonlijke ontwikkeling zelfstandig willen uitvoeren of onder begeleiding. | 59 |