**Table of contents**

[1. Overview 2](#_Toc450127892)

[2. Situation 2](#_Toc450127893)

[3. Task 2](#_Toc450127894)

[4. Action 3](#_Toc450127895)

[5. Result 3](#_Toc450127896)

[6. Reflection 4](#_Toc450127897)

# Overview

<https://meesterschap.wordpress.com/2013/06/09/starrt-methode/>

Good morning everybody, as students of the PXL we were asked to create a platform for the italent course. We will try to give you a brief insight in how we accomplished this task.

* Me, Jesse will first explain how we got into this mess
* Then, Niek will provide you with some details about the task we were facing
* When Niek finishes, Arjen will …

# Situation

As IT students of PXL, we were asked to create a platform for students and teachers to launch ideas for the italent course. This platform will be used to provide students with ideas and stimulate collaboration between departments.

We accomplished most of the technical workload at home. Meetings with clients and coaches were organized in the PXL buildings and team meetings were held using Skype or face to face at OfficeCenter or various other locations.

# Task

Students, teachers and potentially companies can post their ideas on this online platform but that doesn’t mean they have to get involved. The platform is purely an ideas pitcher.

Our task was to create this platform using the latest techniques. Because of this big assignment we had to split our team to get the most of everyone’s experience.

* Arjen did most of the frontend work
* Jesse was dedicated to Hibernate and Spring
* Dennie set up a basic Spring Boot backend and some security
* And Niek tried to resolve technical issues, related tasks and analysis.

Our goal was to create a web-based future-proof and secure Single Page Application with open source libraries and new techniques. We started the creation of the platform by forming a plan. This plan was obviously intended to meet the requirements but we soon realized that the requirements were fuzzier than we first thought. This initial plan is currently still in development, but we decided to go ‘Agile’.

# Action

So, what did we actually do?   
As a first step, we brainstormed a lot. We created an initial exploring analysis and realized that everyone had a different idea on how we should move on. So we decided to arrange some more meeting to discuss how we could meet the goals that were requested and at the same time combine our visions.

Next, we started the deeper analysis process while some of us were looking into the technical details of the application. We tried to get solutions for technical issues we were sure we had to overcome.  
For example: how do we upload videos or pictures? Will we link to them or host them on our website?   
The outcome of these questions were, if course, extremely important for further analysis so we arranged a ‘daily’ scrum meeting using Skype, every day at 8pm sharp.   
Since most of us have a full time job it’s not easy to attend all sessions, so we decided to only attend them if necessary.

When the biggest part of our analysis was done, we started the initial development. Using everyone’s own skills and expertise we managed to get a potentially shippable product in an extremely short matter of time.  
Of course, this product had only about 20pct of all requirements we had to meet.

Today, we are still iterating and resolving requirements and are very excited to show you our final product during the final presentation.

I’d like to give the **FLOOR** to …

# Result

So, the results of all described work was that we had some user stories and requirements done.  
I will provide you with some of these requirements and illustrate the results we have gained.

* Students/teachers should login

This user story is a matter of security and we were determined to accomplish this task future-proof.  
So we introduced spring-boot where security was provided as-is and only needs fine-tuning. Accomplishing this task was more difficult- and took more time than planned but since the entire security of our application came across this task, we were happy to finalize this within 2 weeks.

This task was very urgent since other development tasks depend on the login task. We were able to work further on other user stories but others were put aside.

* Project list should get displayed

If course, a project list should get displayed for multiple people. They should see different results when asking for this list. A guest, for example, can only see ‘public’ projects and students can only view ‘backed’ projects. Backend projects are projects that are verified by teachers.

As you can imagine, this small user story contains a certain grade of complexity and should be discussed as a group.

So we did and came across a solution to create multiple lists to display the information in phase 1, and combine this information into a single list during phase 2, when a user has the possibility to log in.

We learned a lot from everyone’s specific expertise & created an atmosphere were everyone listened to all proposals before deciding which way to go.

# Reflection

Looking back at our goals, we should definitely do things slightly different in future projects.  
But we are proud of our final product and our team’s collaboration.

* Planning

Since we decided to plan everything up-front, we were able to delegate tasks between our team members. We were not very prepared to changing requirements and additional work, so we struggled along these obstacles by re-planning the entire release.

We learned that it’s not a good idea to plan up-from too much because of the changing requirements and unplanned obstacles we had to overcome before other tasks could get completed.  
After some weeks of planning and analysis we decided to go ‘Agile’. We dropped the entire planning and just planned up to the next client meeting. This approach was way better since our requirements to this short amount of time were very clear to everyone & all team members know they had to resolve their tasks before the next sprint could get started.

* Meetings

We held lots of online-meetings. We tend to discuss lots of ‘irrelevant’ information during these meetings since we all wanted to go into detail. These should have been more to the point since development-time gets lost during long meetings where too little is decided.

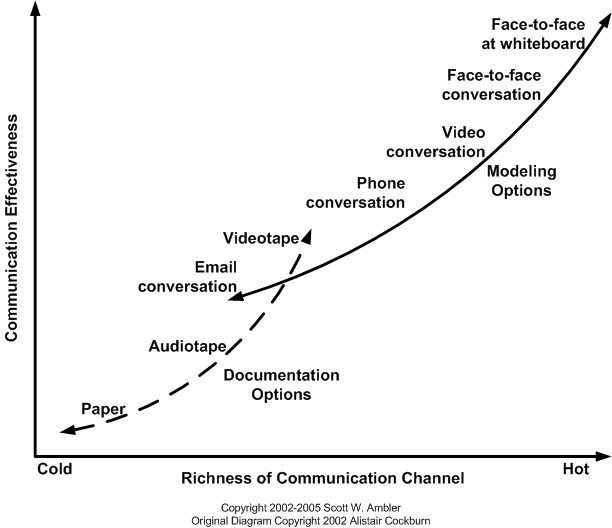
* Deciding the requirements

Requirements are the first step on creating product. It’s impossible to create a product right from the first time. We thought we could, so we created a detailed analysis as documentation of the final product.

Since these requirements changed over time, the ‘final analysis’ got deprecated.   
In future projects, we would only create detailed analysis for requirements that are 90pct sure to implement. Reducing loss of time should be a priority that all team members should take into account.

* Communication

Daily Skype meetings, weekly face-to-face meetings & multiple client meetings were more than enough to get the product we wanted to achieve. Since communication is a key to success for all projects we took this seriously. We not only discussed ongoing problems but also kept a list of features we could implement and improvements on already-developed requirements.



Since face-to-face is the most effective communication, we decided to plan some meetings at OffiCenter. During these meetings, most of the requirements were discussed since they are the foundations of our software. During phone conversations we held every day, we discussed the progress & current issues.

* Tooling

Every developer has his favorite toolset. We wanted to make sure we could use the individual chosen tools. Because, of course, development could go much faster when using tools we know.   
For example, Arjen likes to work with Webstorm, while Jesse prefers Eclipse

# Transfer

So, for future projects we would most certainly keep:

* Daily communication using Skype & weekly face-to-face communication
* Known & supported open source technology with a broad community
* Free and stable build- and deploy processes
* Planning tools

We would most certainly change:

* Getting the requirements clear before further actions
* Don’t plan ahead too much

Things we could do different, but were good enough:

* More face-to-face meetings
* Create less analysis, only when needed