

# Table of contents

1.	Overview.....	2
2.	Situation : JESSE .....	2
3.	Task : JESSE .....	2
4.	Action : ARJEN.....	3
5.	Result : ARJEN .....	4
6.	Reflection : NIEK .....	4
7.	Transfer : DENNIE .....	6

## 1. Overview

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

Good morning everyone, I am Jesse and we are team 1. We are all in our final year of bachelor in IT at the PXL and we were asked to create a team project. In this presentation we will show how we handled things so far. I will tell some more about the project and the tasks we were and still are facing, but let me introduce you to the team first:

Arjen, our front-end specialist, will tell something about the actions we took and the results we've booked.

After that, Niek, our back-end and analysis specialist, will reflect on those results.

And last but not least, Dennie, our security expert will explain how we will take our newly gained knowledge into the future.

## 2. Situation : JESSE

About the project; as I said earlier we are all in our final year and because of that we need to create a project in team. This year the PXL decided to split us up in 4 teams and proposed two different projects to work on. We chose the ITalent project. In the ITalent course students have to spend a certain amount of hours in different domains. The purpose of the project is to create a platform for students and teachers to launch ideas for the ITalent course and to stimulate collaboration between departments.

## 3. Task : JESSE

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. Something like kickstarter, but more transparant and not money-related.

Idea's for projects can be presented with several media like pictures, movie files and presentations.

Students and teachers will be able to like a project and subscribe to a project.

When a project is started there should be a way to view and handle status updates.

Our task is to create this platform using the latest techniques. The task could get split into these sections:

- Communication : team meetings, meetings with client
- Planning : where we have chosen for a 'SCRUM' approach
- Analysis : requirement analysis was the most important one here

- Documentation : Not easy to do as a developer, but quite necessary
- Development : The actual creation of the platform
- Quality Management : Working code, good user experience, etc

Our goal is to create a web-based future-proof and secure Single Page Application with open source libraries and new techniques.

I will now give the floor to Arjen who will tell you about the actions we took and results we made so far.

#### 4. Action : ARJEN

(@ Arjen: je kan hier nog stukken van nemen als je wil. Vond het meer passen bij action dat bij mijn (Jesse) stuk. Of laat het gewoon volledig weg. Dat is ook ok)

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 and PXL buildings.

Since this projects is a quite big one, we've decided to manage the workload using various tools like planning, automated building & deployment tools to speed up development work.

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 and related tasks.

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, so we decided to go Agile.

Of course we didn't strictly follow these steps, one was often mixed with another

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 meetings 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, of 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 excited to show you our final product during the final presentation.

## 5. Result : ARJEN

The result is a **nice-looking responsive webapp** designed in AngularJS and Bootstrap.

The backend is created in Java using Hibernate and Spring.

The advantages of this approach is that it is possible to run the application on any device.

Also, during coding, we could work the 'model-first' approach where we weren't bored with database development. This approach saved us a lot of time since we didn't have to code very much: the focus was coding the processes, not the technical aspects.

The maintainability of the product is very high. There are just a few references to specific classes and no repeated coding we tried to live up to the **High cohesion, loose coupling** paradigm to make sure the quality of our final product was good.

Another result of the implementation of this paradigm is that we have created lots of testcases. This also provides us with quality in coding and maintainability.

Niek will now reflect on all work we have done.

## 6. Reflection : NIEK

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-front 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 knew 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

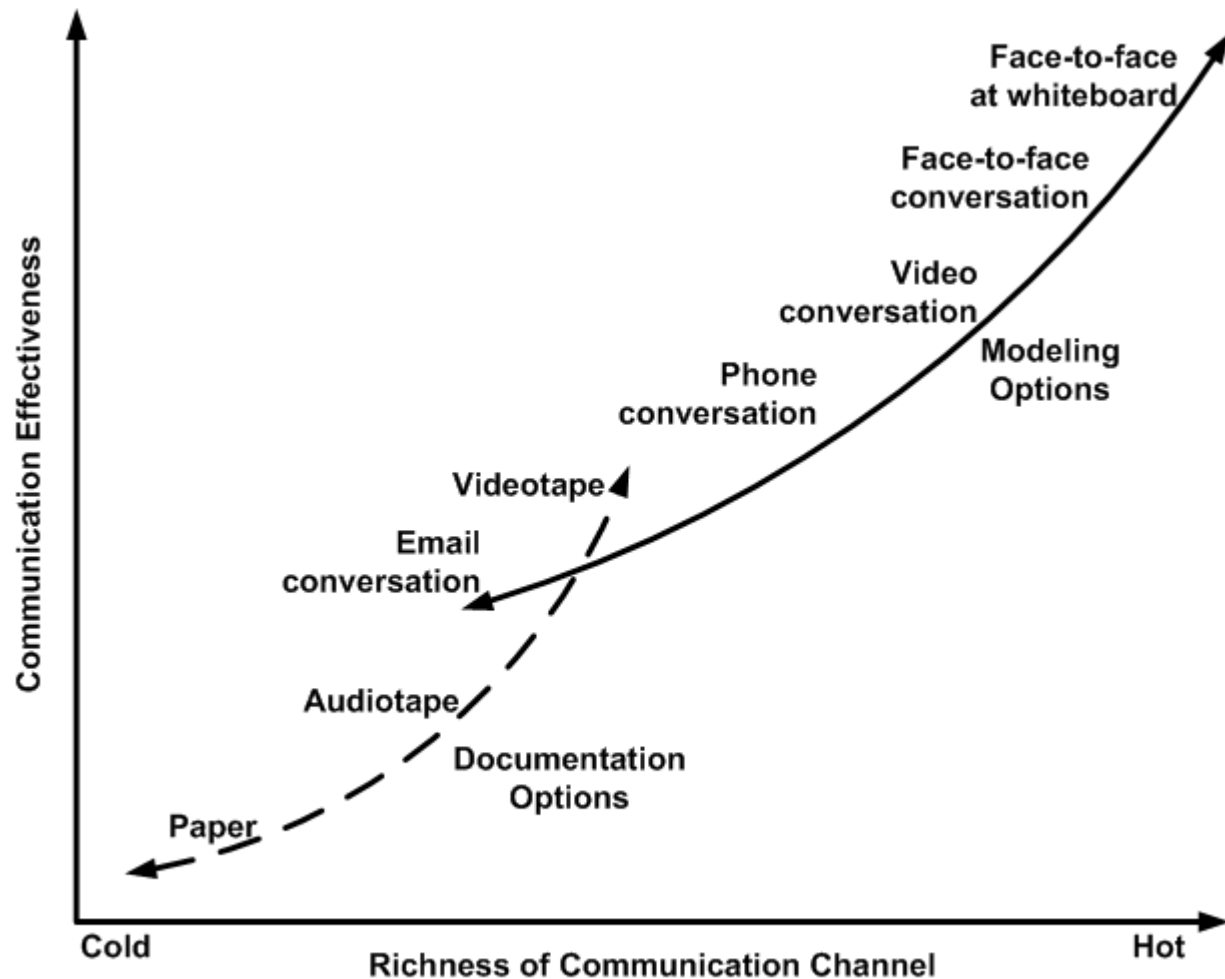
Requirements are the first step on creating product. It's impossible to create a product right from the first time. We thought we could, so we created a detailed analysis as documentation of the final product.

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.



Copyright 2002-2005 Scott W. Ambler  
Original Diagram Copyright 2002 Alistair Cockburn

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 IntelliJ/Webstorm, while Jesse prefers Eclipse

## 7. Transfer : DENNIE

So, for future projects we would most certainly keep:

- Daily communication using Skype & weekly face-to-face communication

Communication is key for collaboration, so it's necessary that all team members are dedicated to communicate and provide critical comments on work so far. That's the only way to accomplish the team's goals.

- Known & supported open source / cross platform technology with a broad community

Choosing the 'right' technology is never easy. We have learned that the open source community provides splendid documentation and that most questions will be answered within hours. This is a big plus in limited-time development.

- Stable build- and deploy processes

Even though most development is done locally on the programmer's own device, using a continuous integration software build and deployment process is a big plus. It is easy to implement and testing can be done on a system that resembles (or is) the production environment, what means that we can be sure that the end product is what we expect it to be. We'd like to take this approach to other projects as well.

- Planning tools

Last but not least, a project always starts with planning ahead. You can keep it basic and use an excel sheet or plain document to provide to your team, but this is a very limited approach. We have chosen for a complete solution from Rational.

This tool provided us with a burndown chart, defect tracking, sprint & release planning and so much more. It's easy to install a great tool for versioning as well.

The tool helped us get on track with planning tasks/defects and team member's efforts.

We are convinced that a good planning tool can make or break a project's deadline.

### **We would most certainly change:**

- Getting the requirements clear before further actions

New IT projects should go the 'Agile' way. But we have learned that it all starts from requirements. When the requirements are not clear, there is no point in creating an analysis.

We will definitely take this into consideration since we have lost lots of time in analyzing requirements that became deprecated or had to change for one reason or another

- Establish early on what technology to use

We already mentioned that we would continue using Open Source and Cross Platform technology, but an important side note to this is that we should finalize early on what technology that should be. Even minor changes of in the base of your project could lead to grave changes in the rest of the project.

We experienced this when we decided to change from Spring to Spring Boot, which rendered most of the currently written back end code unusable.

- Don't plan ahead too much: create time for a retrospective

Planning and agile go hand in hand, but only if you do not plan too much in advance. The planning should be flexible enough to re-locate work but tight enough to keep the team going.

Our planning was way to tight. We didn't have lots of time to reflect after some time and as a result we had to change some requirements in a later stage. It's important to reflect after each sprint so changes in requirements are quickly determined.

### **Things we could do different, but were good enough:**

- More face-to-face meetings

Face to face communication between team members gets the most results. Just one a week or one each two weeks is the bare minimum to get the project done in time, but make sure not organize too many meetings that don't add anything substantial.

Phone calls also work, but they are not that effective since you miss a lot of aspects that face to face does provide, for example, body language of your team members.

- Create less analysis, only when needed

To conclude, analysis is necessary, but we found out that too much isn't perfect in a project as the one we're currently in. We're very limited in time; it is very time consuming and lot of it ends up in a box or on a drive and never sees the light of day again, simply because the nature of the project dictates that our analysis will change. We've learned that it's better to create analysis when it's needed.



I would like to thank you all for your attention

Questions??????????