

University College of Northern Denmark

Technology and Business

Computer Science Academy Profession (AP) Degree

Class: DMAJ0916

Title: System Development – 3rd Semester 2017

Project participants (Group 2):

-Ralfs Zangis

-Andrei-Eugen Birta

-Hannes Heiskonen

-Stoycho Nenov Anastasov

Supervisors:

-Brian Hvarregaard

-Per Trosborg

Due date: 2017-December-18 (13:45 UTC+1)

Submission date: xxxx

Birta Zanigis Heiskonen Nenov

Contents

[1. Preliminary Study 4](#_Toc499488192)

[1.1. Introduction 4](#_Toc499488193)

[a. Project Idea 4](#_Toc499488194)

[b. Problem Statement 4](#_Toc499488195)

[1.2. Plan driven Vs. Agile Development (elaborated through methods) 4](#_Toc499488196)

[1.2.1. Plan Driven 4](#_Toc499488197)

[a. Waterfall 4](#_Toc499488198)

[1. Main Idea 4](#_Toc499488199)

[2. Pros 4](#_Toc499488200)

[3. Cons 4](#_Toc499488201)

[b. Unified Process 4](#_Toc499488202)

[1. Main Idea 4](#_Toc499488203)

[2. Pros 4](#_Toc499488204)

[3. Cons 4](#_Toc499488205)

[1.2.2. Agile 4](#_Toc499488206)

[a. SCRUM 4](#_Toc499488207)

[1. Main Idea 4](#_Toc499488208)

[2. Pros 4](#_Toc499488209)

[3. Cons 4](#_Toc499488210)

[b. Kanban 4](#_Toc499488211)

[1. Main Idea 4](#_Toc499488212)

[2. Pros 4](#_Toc499488213)

[3. Cons 4](#_Toc499488214)

[1.3. Quality Assurance 4](#_Toc499488215)

[a. Planning Principles 4](#_Toc499488216)

[b. Development Principles 4](#_Toc499488217)

[1.4. Quality Criteria and Architecture 4](#_Toc499488218)

[2. Development Process 4](#_Toc499488219)

[2.1. Development Method of Choice 4](#_Toc499488220)

[a. Final Choice 4](#_Toc499488221)

[b. Selected Criteria 4](#_Toc499488222)

[c. Pros 4](#_Toc499488223)

[d. Cons 4](#_Toc499488224)

[2.2. Sprints Summary 5](#_Toc499488225)

[2.2.1. Sprint 0 5](#_Toc499488226)

[a. Sprint Backlog 5](#_Toc499488227)

[b. Burndown Chart 5](#_Toc499488228)

[2.2.2. Sprint 1 5](#_Toc499488229)

[a. Sprint Backlog 5](#_Toc499488230)

[b. Burndown Chart 5](#_Toc499488231)

[2.2.3. Sprint 2 5](#_Toc499488232)

[a. Sprint Backlog 5](#_Toc499488233)

[b. Burndown Chart 5](#_Toc499488234)

[2.2.4. Sprint 3 5](#_Toc499488235)

[a. Sprint Backlog 5](#_Toc499488236)

[b. Burndown Chart 5](#_Toc499488237)

[2.2.5. Sprint 4 5](#_Toc499488238)

[a. Sprint Backlog 5](#_Toc499488239)

[b. Burndown Chart 5](#_Toc499488240)

[3. Epilogue 5](#_Toc499488241)

[3.1. Conclusion 5](#_Toc499488242)

[3.2. References 5](#_Toc499488243)

# Preliminary Study

## Introduction

This document summarizes the collaboration of Group 2 for the system development exam of the 3rd Semester. The group consists of 4 members of 4 different nationalities. Despite the major differences in our opinions, we agreed on certain rules and guidelines to follow, thoroughly elaborated in the accompanying document called group contract; we managed to harness the

benefits of diverse ideas and identify multiple possible approaches to certain problems.

### Project Idea

The idea of the project is to create a service that handles multiple chatrooms with limited people count capacity, where people can discuss topics they are passionate about; which the users may access either via the web client or the windows application. Listening to music through the application using YouTube Data API, joining as groups of people or simply playing a game of rock-paper-scissors, are other major user stories of our program.

### Problem Statement

Some of the problems we had to find the answer to include, but are not limited to: Users whose number is greater than the available slots in a given chatroom try to join at the same time (for example, there is 1 slot left but 2 people try to join), a group whose number of people is greater than the available slots in a given chatroom tries to join (for example, a group of 4 attempts to join when there are only 3 or less slots left), preventing SQL Injection, learning how to use Callbacks and bindings.

After formulating the problem statement and having it approved by the supervisors we were assigned the task of solving the problem by selecting the most suitable agile system development method based on the situation, through well-planned and well-synchronized teamwork.

The following portion of the report represents the theoretical part.

## Plan driven Vs. Agile Development (elaborated through methods)

### Plan Driven

#### Waterfall

##### 1. Main Idea

##### 2. Pros

##### 3. Cons

#### b. Unified Process

##### 1. Main Idea

##### 2. Pros

##### 3. Cons

### 1.2.2. Agile

#### a. SCRUM

##### Main Idea

##### Pros

##### Cons

#### b. Kanban

##### Main Idea

##### Pros

##### Cons

## Quality Assurance

Quality assurance is part of project which is making sure that in the end, a high quality product, that satisfies the customer’s requirements, will emerge. Just as the old English proverb says: “better safe than sorry”, assuring the quality of a product, before and during the development process, is usually a cheaper and faster process of preventing errors than fixing them in a sloppy way, at the last moment.

Some external quality attributes include maintainability, reusability, reliability and usability. External quality can be ensured with pair programming, test driven development, continuous delivery and unit testing.

There are several ways to ensure the quality of a product. Following chosen development method’s instructions is first step towards developing high quality product. Final step is to have quality control. Quality control deals with testing already existing product. People in quality control make sure that the end product came out exactly as planned and the product fulfils all the requirements.

## 

## Quality Criteria and Architecture

# Development Process

The following portion represent the practical part of the report and is meant to show how exactly did we apply the theory, in our work.

## Development Method of Choice

Although we have learned about the existence of many development methods, each with its own unique characteristics, we decided that the best way of choosing a development method is by evaluating the team and creating a Boehm and Turner Model.

The following image shows the model we have ended up with and according to which, we have chosen a development method.



### Final Choice

From the diagram above, resulted that we needed some kind of agile method, due to the high amount of expected changes and Critical level of the product, that is structured enough in order to accommodate for the high number of low level developers and Culture. We have decided to use a combination of the SCRUM and the XP methods of development taking daily meetings (usually at 10 a.m. at school) and sprint structure of the project work from SCRUM while ensuring the robustness of our program through XP and/or pair programming if necessary.

In the end we have observed that the selected criteria, came with the following perks:

### Pros

* The daily meetings allow to very well understand the stage at which the project is as well as identify any possibly issues and setbacks as soon as possible before they become a major problem.
* Splitting the work into sprints provides the opportunity to evaluate how well-planned was each iteration and make changes if necessary (learn from our mistakes). For example, if we assign too many tasks and fail to accomplish them by the end of the sprint, for the next sprint we will be able to adjust and plan correctly.
* Quickly identify and solve any problems that were lurking and appeared not to be there.
* Pair programming can make different developers more familiar with the code overall rather than just their own part.
* Optimizing and testing the code all the time greatly reduces the chance of having major bugs or missing functionality.

### Cons

• Planning meetings and travelling daily takes some time, which could’ve been used on work instead.

• The product owner can change their mind at any point.

• Pair programming can sometimes take more time than wanted since 2 people are focusing on the same task.

## Sprints Summary

Sprint work was mainly organized by scrum master and decided by product owner. Each working day started with a short meeting, in the class room or other location previously decided by group (it usually happened at 10:00 unless majority of group agreed for different time, according to the group contract), where group members, with help of the scrum master, discussed the following things- which goals have been achieved since last meeting, what goals are set by the next meeting and if any problems were encountered so far.

After the meeting, work was started on the project, if there were any problems, the scrum master or any other person with experience would help bring a resolve to it. On the last sprint day there was sprint review where group members discussed the good and bad about this sprint and how can we improve in these fields, followed by goal setting for next sprint and work distribution. This process resulted in our estimations getting progressively better, thus allowing us to set realistic goals.

In the following portion we are going to give you a detailed diary of what we have done each sprint.

### Sprint 0

This sprint’s roles were distributed as follows:

Product owner- Hannes Heiskonen

Scrum master- Ralf Zangis

One of the first things we needed to establish were the user stories, and once they were established we needed a way to organize them depending on their importance. And the best way to do so, is with the help of a MoSCoW Model, so the following, colorful, table, shows our final version of it.

|  |  |  |  |
| --- | --- | --- | --- |
| Must | Should | Could | Wont |
| ------------- | ------------- | ------------- | ------------- |
| Register | Forgot password | Show if someone is writing | Follow/subscribe |
| Login/Logout | Multiple chats for user | Multiple chats for person | Friends |
| Manage profile | Invite to chat | Manage YouTube playlist | News area |
| Manage public chat | Show online users | Multiplayer game |  |
| Manage private chat | YouTube video player | Join with group |  |
| Manage messages | User can’t log in if he is online |  |  |
| Dedicated client | Manage group |  |  |
|  | Web client |  |  |
|  |  |  |  |

The first version of the model looked quite differently, having a lot of text in the “Must” column and nothing in the last two columns. But as time passed and after we pivoted from the original idea, a few times, the final version of the model slowly started to be appear, resulting in its concluding form sometime in the middle of Sprint 1.

#### Sprint Backlog

The following table represents the last version of our Sprint’s 0 backlog and the accompanying image represents the burndown chart.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Priority | Work | time-est. | time-act. | Person |
| 2 | Mock up | 4 | 5 | Ralfs |
| 2 | MoSCoW | 3 | 3 | Ralfs |
| 1 | Use cases | 2 | 3 | Ralfs |
| 2 | Distribute use cases | 1 | 2 | Ralfs |
| 1 | Group Contract | 9 | 9 | Andrei |
| 2 | Domain model | 7 | 5 | All of us |
| 3 | Domain model explanation | 2 | 2 | Stoycho |
| 2 | Database model | 12 | 13 | All of us |
| 3 | Database table creation scripts | 2 | 2 | Andrei |
| 3 | Drop database script | 1 | 1 | Andrei |
| 3 | Database trigger tests | 2 | 2 | Andrei |
| 3 | Database insert row scripts script | 3 | 3 | Andrei |
| 1 | Backlog | 7 | 8 | All of us |
| 1 | User stories | 10 | 11 | Hannes |
| 3 | Create report structure | 3 | 3 | Hannes |
| 1 | Generate product idea | 10 | 14 | All of us |
| 3 | Spike on how chats work | 3 | 4 | Ralfs |
| 1 | Decide on development method | 2 | 2 | All of us |
| 2 | Establish coding standards | 3 | 2 | All of us |
| 2 | Agree on working conditions | 2 | 2 | All of us |

Although the sprint was 14 days, we have taken into account only the 5 days that, represented by the bottom set of numbers in the chart, in which we have spent the most time working on the project.

### Sprint 1

This sprint the assigned roles have been changed to the following:

Product owner- Ralf Zangis

Scrum master- Stoycho Anastasov Nenov

Sprint 1 was possibly the most important regarding the program, as in this sprint we made multiple spikes on new technologies like binding, WCF, YouTube API, different communication protocols and email sending. It resulted in the main features of the project being finished, besides that sprint also tested the team, as there we had conflicting arguments regarding future of project, which was solved by product owner and scrum master agreeing on middle ground, which resulted in project main idea changing drastically while keeping main user stories.

The following table represents this sprint’s backlog and the accompanying image is the burndown chart.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Priority | Work | time-est. | time-act. | Person |
| 1 | Register | 3 | 5 | Andrei |
| 1 | Login | 2 | 3 | Andrei |
| 2 | Forgot password | 3 | 3 | Andrei |
| 1 | Get profile | 3 | 3 | Andrei |
| 1 | Update profile | 3 | 2 | Andrei |
| 1 | Delete profile | 2 | 2 | Andrei |
| 2 | Spike on email sending | 3 | 2 | Andrei |
| 1 | Multiple chats for one client | 2 | 1 | Ralfs |
| 1 | Create Chat | 3 | 3 | Ralfs |
| 1 | Update Chat | 3 | 3 | Ralfs |
| 1 | Find chat | 2 | 2 | Ralfs |
| 1 | Delete Chat | 2 | 2 | Ralfs |
| 1 | Join chat | 4 | 4 | Ralfs |
| 1 | Leave chat | 2 | 2 | Ralfs |
| 1 | Send message | 5 | 4 | Ralfs |
| 1 | Receive Message | 4 | 4 | Ralfs |
| 1 | Get messages | 3 | 3 | Ralfs |
| 2 | Show message info | 3 | 2 | Ralfs |
| 1 | Remove your message | 2 | 3 | Ralfs |
| 1 | Spike on different communication protocols | 1 | 1 | Ralfs |
| 2 | Close chat if its deleted | 2 | 2 | Ralfs |
| 1 | Spike on WCF | 2 | 2 | Ralfs |
| 2 | Invite person to chat | 4 | 4 | Ralfs |
| 2 | Get notification | 3 | 4 | Ralfs |
| 2 | Remove notifications | 2 | 2 | Ralfs |
| 1 | Spike on binding | 6 | 10 | Ralfs |
| 2 | Get songs info | 4 | 5 | Stoycho |
| 1 | Spike on youtube | 6 | 6 | Stoycho |
| 1 | Fix problem statement | 2 | 2 | Hannes |
| 2 | Choice of method | 4 | 4 | Hannes |
| 1 | Plan driven vs agile | 3 | 4 | Hannes |
| 1 | Dedicated client | 25 | 30 | All of us |

Although the sprint tested our group’s work limits, it was still a great success because we could agree on common project path and it resulted in all necessary features being successfully implemented.

In the last day of the sprint, we had a meeting with all the other classmates, where we presented a working product and talked about the backlog. After that, just as a professional, conscious chef, we prepared for the next sprint, by assigning the new roles and putting together a sprint backlog.

### 2.2.3. Sprint 2

This sprint the assigned roles have been changed to the following:

Product owner- Andrei-Eugen Birta

Scrum master- Hannes Heiskonen

This sprint’s main purpose was finishing touches for the existing features, such as optimizing, refactoring, fixing some of the most obvious “features” and just adding a little sprinkle on top, to make it shine from a crowd of other projects. Because of this sprint, we had a completely working dedicated client with all the intended features.

Since this sprint’s goals were achieved much faster than anticipated, we started working on the second client, that being the web client and web server.

The following table represents this sprint’s backlog and the accompanying image is the burndown chart.

| Priority | Work | time-est. | time-act. | Person |
| --- | --- | --- | --- | --- |
| 1 | Play song | 1 | 2 | Stoycho |
| 1 | Save song | 1 | 1 | Stoycho |
| 1 | Create playlist | 2 | 3 | Stoycho |
| 1 | Get playlist | 1 | 1 | Stoycho |
| 2 | Update playlist | 2 | 2 | Stoycho |
| 2 | Delete playlist | 1 | 1 | Stoycho |
| 2 | Multiplayer game | 6 | 5 | Andrei |
| 2 | Game debug/release mode execution error | 3 | 10 | Andrei |
| 1 | One person can be logged in at once | 2 | 2 | Andrei |
| 1 | Logout | 2 | 2 | Andrei |
| 1 | Refactor database | 4 | 6 | Andrei |
| 1 | Join as group | 10 | 8 | Ralfs |
| 2 | burn down chart | 1 | 1 | Ralfs |
| 2 | Updating end expanding existing UI | 1 | 2 | Ralfs |
| 1 | Join group | 1 | 1 | Hannes |
| 1 | Leave group | 1 | 1 | Hannes |
| 1 | Create group | 1 | 1 | Hannes |
| 1 | Read group | 1 | 1 | Hannes |
| 1 | Update group | 1 | 1 | Hannes |
| 1 | Delete group | 1 | 1 | Hannes |
| 2 | Exception handling | 5 | 3 | Andrei |
| 2 | Exception handling | 4 | 3 | Ralfs |
| 2 | Exception handling | 2 | 2 | Stoycho |
| 2 | Exception handling | 2 | 1 | Hannes |
| 2 | Report planing | 20 | 24 | All of us together |
| 2 | Spreading tasks | 8 | 8 | All of us together |
| 2 | Chat users updated if user info changed | 1 | 1 | Ralfs |
| 2 | Show if someone is writing | 1 | 1 | Ralfs |
| 2 | Show online persons | 2 | 2 | Ralfs |
| 2 | Show newest chat info | 3 | 2 | Ralfs |

This sprint was possibly the most “exciting” of them all, simply because of the problems we have encountered, such as both the Video Player and Rock-Paper-Scissors game, working only in Debug Mode.

### 2.2.4. Sprint 3

This sprint the assigned roles have been changed to the following:

Product owner- Stoycho Anastasov Nenov

Scrum master- Andrei-Eugen Birta

We like to call this Sprint as “The beginning of the end”, simply because this is the sprint in which we intended to wrap up the project, at least start doing so. The main goals for this sprint were finishing the web client, finishing touches for the service itself and started writing the reports. And as usual, all of this sprint’s tasks can be seen in the following table, sprint backlog and the accompanying image, burndown chart.

#### a. Sprint Backlog

#### b. Burndown Chart

But of course, since we have never previously worked with MVC, fell right into Mr. Brian’s “prediction”. Just as you can see from the burndown chart, this sprint’s estimations were just as close to reality as we would ever get to winning the lottery. We had to do a spike on MVC, SignalR and several other web client related technologies.

Although at the begging of the sprint we were slightly ahead of schedule, at the end of it, time became an enemy and the race to handing in a “finished” product, began. The only good part of this sprint being that our knowledge slightly increased, regarding web clients.

### 2.2.5. Sprint 4

#### a. Sprint Backlog

#### b. Burndown Chart

# 3. Epilogue

## 3.1. Conclusion

In conclusion, during this semester’s exam we have managed to achieve not only knowledge about different frameworks of developing software, such as Extreme Programming, SCRUM and Kanban, and seeing how actual developers work, through the company visits; but also how to communicate and reach out to companies of the profile.

As a ending note, we would like to thank all of the viewers, who invested their time in reading this paper and also to the guiding teachers that helped and guided us through the entire process.

## 3.2. References