**SCRUM PLANNIG**

Zinema – Management Software

Distributed System

**Students**

Andrei Cioanca – 266105

Claudiu Rediu – 266129

Dominika Kubicz – 266148

Nikita Roskovs – 266900

Stefan Harabagiu – 266116

**Supervisors**

Jakob Knop Rasmussen

Jan Otto Munch Pedersen

**Contents**

[**1.** **SCRUM ROLES** 3](#_Toc532721979)

[**2.** **Sprint Planning** 3](#_Toc532721980)

[**3.** **Group Meetings** 3](#_Toc532721981)

[**4.** **Sprint Backlogs** 4](#_Toc532721982)

[**5.1 Burndown Chart** 9](#_Toc532721983)

# **SCRUM ROLES**

SCRUM Master: Claudiu Rediu

Product Owner: Dominika Kubicz

# **Sprint Planning**

1st Sprint:

8th of October – 12th of October

2nd Sprint:

22nd of October – 16th of October

3rd Sprint:

29th of October – 2nd of November

4th Sprint:

5th of November – 9th of November

5th Sprint:

12th of November – 16th of November

6th Sprint:

19th of November – 23rd of November

7th Sprint:

26th of November – 30th of November

8th Sprint:

3rd of December – 7th of December

9th Sprint:

10th of December – 14th of December

# **Group Meetings**

Group meetings will be on Fridays. They will consist of SCRUM Review and Retrospective.

They will all be held at the university. There will be approximately 150 hours of work each sprint.

# **Sprint Backlogs**

1st Sprint

Sprint Review:

Work was split in multiple days as immediate priorities took held of our schedules. The meetings got heated because of arguments, but it didn’t impend the work pace. Due to planning, everything was achieved even before the term. The results of designing the system were satisfactory.

Sprint Retrospective:

WHAT WENT WELL:  
 - TEAMWORK

- WORK PACE

- FOCUS

WHAT COULD BE IMPROVED:

- COMMUNICATION

The way we planned our schedules worked properly to satisfy both personal and professional. Work pace and focus were centered on the work, but communication got heated because of arguments. They were solved but could’ve been avoided by a bit more tact. In future meetings situations like these will be dealt with a better approach.

2nd Sprint

Sprint Review:

The design is still the focus. Meetings took place to prepare for future sprints in which we will implement the system. The schedule for meetings was unaltered by external issues. In this sprint, the process evolved further in designing the system.

Sprint Retrospective:

WHAT WENT WELL:  
 - TEAMWORK

- WORK PACE

- COMMUNICATION

WHAT COULD BE IMPROVED:

- FOCUS

In comparison with the previous sprint, communication was on point. Arguments were dealt with a better attitude, especially openness. Focus started to fall off towards the end, but it didn’t affect the result as most of the work was already done. The working medium was a better environment for working and discussing.

3rd Sprint

Sprint Review:

The focus was designing the proof of concept. The common agreement was to work on the first component and have it working as intended, in order to demonstrate the feasibility of our system. The construction of the system also started with this sprint

Sprint Retrospective:

WHAT WENT WELL:

* TEAMWORK
* COMMUNICATION
* FOCUS

WHAT COULD BE IMPROVED:

* UNDERSTANDING THE REQUIREMENTS OF THE SYSTEM

As the designing and implementation began, doubts started to appear. This was mainly the only issue with this sprint. There is a need for a meeting with the supervisors to understand in a clearer manner what is expected. The working environment was satisfying in this sprint.

4th Sprint

(Add supervisor meeting with Jan during this sprint. He explained what designing our own protocols means and we remade our code using sockets)

Sprint Review:

After discussing with one of the supervisors and meeting to reflect on the evolution of the code, the design was improved as there was a lack in understanding of the requirements. This was one of the biggest steps until now for meeting the requirements.

Sprint Retrospective:

WHAT WENT WELL:

* ADDAPTIVENESS OF THE TEAM

WHAT COULD BE IMPROVED:

* COMMUNICATION
* TEAMWORK
* FOCUS
* WORK PACE

In comparison with previous sprints, the results of planning and splitting the tasks was not fruitful. It was not such a big issue as most of the code was scrapped. Looking forward to future sprints, plans should be made in such a way that the direction is clear and easy to follow.

5th Sprint

(Add supervisor meetings with Jakob and Jan. They both explained how we didn’t really respect the 3-tier model and would be a good idea to have the database server separate. We decided after this to code it in C# and communicate over sockets.)

Sprint Review:

The proof of concept created was reviewed by the supervisors. After the meeting, design was adjusted and tasks on how to improve it were done. Requirements were made clearer and the vision for the system is becoming clearer.

Sprint Retrospective:

WHAT WENT WELL:

-COMMUNICATION

-TEAMWORK

WHAT COULD BE IMPROVED:

-FOCUS

-WORK PACE

Comparing it to previous sprints, the vision for the final product became more defined. The pace should be improved as the number of sprints until the hand in are decreasing. The problem of members not communicating when in doubt started to become less of an issue. Use of tools like Github and Trello has improved.

6th Sprint

Sprint Review: The project advanced towards the most relevant components. The design now is complete for all components and the only thing to do is finish up on the implementation and finalize the details on the code made in previous sprints.

Sprint Retrospective:

WHAT WENT WELL:

-WORK PACE

-TEAMWORK

WHAT COULD BE IMPROVED:

-FOCUS

This was one of the better sprints in which a lot was achieved. The only thing that could be improved is bringing together in a better form each other’s working habits.

7th Sprint

Sprint Review:

Work on the final component has started. The pace is slower than expected, but there is enough time until the deadline to have everything done. The visual part on each component is almost done.

Sprint Retrospective:

WHAT WENT WELL:

-COMMUNICATION

WHAT COULD BE IMPROVED:

-WORKPACE

-FOCUS

In comparison with other sprints, there hasn’t been much done in the way of code. The focus is slowly shifting towards documentation in the closing sprints. Two more sprints should be more than enough to fulfill the requirements.

8th Sprint

Sprint Review:

The final component is finished. The part with advertisements has been removed from the plan, as it didn’t help in achieving the goals and took too much time for how much it was worth. Documentation and testing is the only focus now.

Sprint Retrospective:

WHAT WENT WELL:

-COMMUNICATION

-WORKPACE

-FOCUS

-TEAMWORK

WHAT COULD BE IMPROVED:

Putting into perspective the other sprints, this was the best one. The implementation has been finalized and splitting the work is going well. Teamwork and communication was great.

9th Sprint

Sprint Review:

All the requirements are achieved. The project was kept simple to ensure that everything worked as expected. Proofing everything that is already done and updating the documentation is the current strategy.  
  
Sprint Retrospective:

WHAT WENT WELL:

-COMMUNICATION

-WORKPACE

-FOCUS

-TEAMWORK

WHAT COULD BE IMPROVED:

This was also a great sprint by the general standards that were created for them. There weren’t any complaints and the tasks were achieved in due time.

## **4.1 Burndown Chart**

The burndown chart’s axes take into calculation the number of tasks and the sprint number. The first sprints were characterized by a focus on setting up the environment for the project and the initial documentation. Over the course of the next sprints, there were external projects and concerns that were not part of SEP that influenced the departure from the ideal tasks on each sprint. During the last sprints, work was done in the ideal manner. It helped in fulfilling the requirements on time and making the most helpful decisions.

# **5. Unified Process**

## **Inception**

During the Inception phase, focus was placed on preparing the working environment and formulating a scope of the project. Working together with SCRUM, this phase was mostly in the first sprints. It consisted of creating the project description, sprint planning and creating the product backlog. Project description was created to formulate a scope, introduce boundaries, a vision for the system and gain acceptance and estimate what the product should end up like. The requirements and the first use cases are created to guide the development of the system. SCRUM was used in this phase to estimate a schedule and delimit what the system should and should not do using the product backlog.

## **Elaboration**

During the Elaboration phase, the main activities were defining and refining the vision and architecture of the system, making sure that the risks are mitigated, ensuring the project is worth further advancement and planning for the construction phase. Possible risks are that the system is not maintainable, understandable or not respecting the requirements. The use cases were created to demonstrate that the system will support the requirements. It was ensured that they would reflect the system that will be developed. Elaboration phase is mostly characterized by the analysis and design of the system.

## **Construction**

During the Construction phase, the priority is constructing the system as rapidly and efficient as possible while having the analysis, design, implementation and testing complete. As each component is implemented, the system is proofed against the requirements and specifications then updated accordingly. Emphasis is put on the working process and its efficiency. The approach was that after each component was implemented, the system could be on its own a product that would work. In this way, the focus was to create it in a way that makes it maintainable and open to the adding of components with new features.

## **Transition**

During the Transition phase, the main goal was that documentation, the user guide and have it ready to present to the customer. By the end of this phase, bug fixing, and enhancements made to the system must be complete. User feedback should be considered for fine tuning the product. The system must be presented to the customer.

## **Distribution between effort and schedule**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Inception | Elaboration | Construction | Transition |
| Effort | 10% | 20% | 65% | 5% |
| Schedule | 11% | 22% | 56% | 11% |

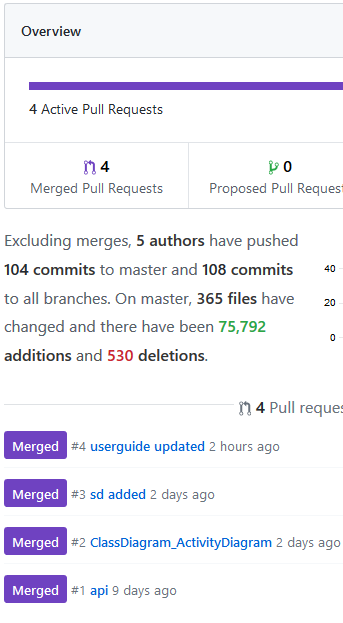
# Tools

## **Git & GitHub**

Git is a revision control system. A tool to manage the source code history of a project.

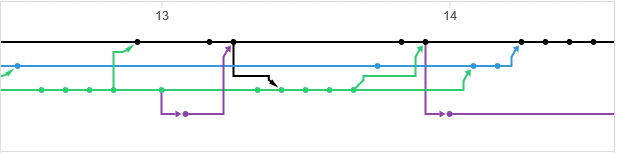
GitHub is the hosting service that was used for the Git repository created for the current project.

There were both used to ensure that the work was centralized.



This is an example of the use of the pull request feature. A pull request announces that there are changes to be pushed, but before that they need to be checked and discussed.

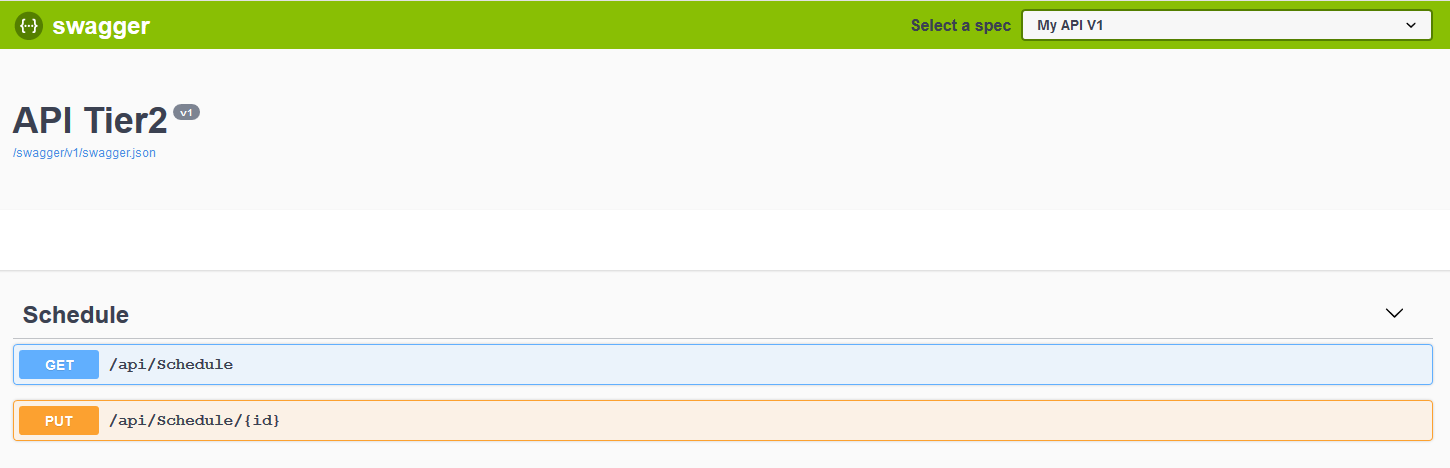
Another feature that was used and is worth mentioning is the branching inside a repository



This is some branches inside the project looked around the 13th-14th of December. They helped in the management of the source code and introduction of bug fixes.

## **Swagger**

Swagger is a tool that helps in designing, building, documenting, testing and standardizing an API. In the context of this project, it was used to document the API.

As soon as the API is started, the documentation can be checked through accessing a certain link(https://localhost:5003/swagger/index.html) 

The actions that can be performed on the API are documented together with the Models.