Streamster

Process Report

Michał Karol Pompa, 266494

Michaela Golhova, 266099

Matej Michalek, 266827

Supervisor

Behnam Boujarzadeh

22601 characters

Software Engineering

7th semester

December 2020

Table of content

1 Introduction 1

2 Group Description 2

3 Project Initiation 4

3.1 Risk assessments 4

3.2 Group contract 5

3.3 Project Description 5

4 Project execution 6

4.1 Unified Processes 6

4.1.1 Inception Phase 6

4.1.2 Elaboration Phase 7

4.1.3 Construction Phase 8

4.1.4 Transition Phase 9

4.2 SCRUM 10

4.2.1 SCRUM roles 10

4.2.2 Working methodology 10

4.2.2.1 Event Storming 13

4.2.3 Burndown chart 14

4.3 Technical tools 17

4.4 List of tasks and responsibilities 17

5 Personal reflections 18

5.1 Michaela Golhova 18

5.2 Michal Pompa 19

5.3 Matej Michalek 20

6 Conclusion 21

7 References 22

8 Appendices 23

1 Introduction

The Bachelor project started at the beginning of the 6th semester. First, the idea of the project was made, and the group decided to create a video streaming application. In the beginning of the 7th semester it was needed to formulate the purpose of this platform, where the group decided to create this application for students and teachers as an alternative way of education due to the pandemic period that hit the entire world. The idea of the project is that teachers can create various groups on the platform, like classes, where they share videos of their lectures and students can access them. Moreover, the platform should help students with their studies and recommend them videos based on their preferences.

2 Group Description

The group has 3 members: Matej Michalek, Michaela Golhova, and Michal Pompa that were classmates and in the same semester groups together since the first semester. That brings great benefits of knowing each other, which makes the work easier and faster. The members from this group come from Slovakia and Poland, and as the cultural background is not so different it was easier to understand each other from the beginning.

Michal Pompa was more experienced in programming at the beginning of studies than the rest of the group, as he was studying programming in high school in Wroclaw and had an internship before the University that helped him to become a leader of the group. Moreover, he was pushing the rest of the team members further as his demands were always high. During the studies, he found a passion for system architecture that he started to pursue by reading books about domain-driven-design, event storming, clean code, and more. Further, he got an internship in Systematic that brought him a lot of experience in system development. He already got a full-time position in Systematic, which he starts in February 2021, right after Graduation.

Matej Michalek started his programming journey at VIA University. As he has great logical thinking abilities from his passion for chess, it wasn’t a problem for him to understand the programming principles. He is a quick learner and has a high demand for details and finishing the job, which brought to group solid results. His preferable choices became working with Data and Backend that he is applying in his work for Slovak company Archiles, where he started to work after his internship in Vienna. He is eager to learn more, that’s why he decided to continue for master’s studies. In his free time, he plays and attends chess tournaments which he started to attend when he was 7 years old. Moreover, he likes to watch sports and he also likes snowboarding during winter.

Michaela Golhova is a passionate Mobile Developer from Slovakia. It is at the VIA University where she has discovered her interests in programming and designing UI for mobile apps. She often spends her free time on learning new mobile development technologies or UI designs principles. During her internship in DIS she was involved in various mobile apps projects. Thanks to that experience, she has secured a full-time position in a company called Danalock as an Android Developer. In her free time, Michaela likes to play various instruments, such as piano, guitar and ukulele or read books and play video games.

3 Project Initiation

3.1 Risk assessments

The following table represents possible risks with their likelihood and severity. Each risk has defined a countermeasure action that is going to be executed when the risk happens.

DESCRIPTION	LIKELIHOOD (1-5)[1]	SEVERITY (1-5)	RISK MITIGATION
Pandemic virus / sickness	4	3	Work from home, communicate through social networks
Lack of knowledge	3	4	Research, study, ask teachers for help
Technical breakdown	2	4	Reallocate workload until faulty equipment is fixed
Not meeting deadlines / milestones	2	5	Always reserve few days for checking and reviews before real deadline
Inaccurate estimations	3	4	Make realistic goals for each sprint and task in particular Use daily stand-ups to communicate about possible delays and redistribute work appropriately or in serious case change sprint's goal

Table 1 - Risk assessment table

[1] Scale: 1 – Low, 5 - High

3.2 Group contract

To avoid conflicts and misunderstanding within a group, the group contract was signed in the beginning of the project and can be found in the Appendix A.

3.3 Project Description

The first version of Project Description was made during the 6th semester. This version was focused mainly on two target groups - families and content creators, as it was discovered that these groups were not satisfied with current video streaming solutions. However, in the beginning of the 7th semester, the Project Description had to be updated because of the change of target groups. The corona pandemic affected many areas of life where education was one of them. On the one hand, the learning process became harder for students which had to find alternative study materials online. On the other hand, teachers had to adapt to new needs of teaching online which often consisted of creating educational videos for students. Since, both sides of education, teachers and students were highly affected by actual corona pandemic, it was decided to consider them as the main target groups and therefore the Project Description was updated. The actual Project description can be found in the Appendix B.

4 Project execution

4.1 Unified Processes

To be able to finish the project on time we created a work structure and schedule using Unified processes (UP) and Scrum work methodology as in previous semesters. Thanks to UP we had an overview of all our tasks, features and their deadlines that help us to keep up with the schedule to accomplish the desired result. (Osis and Donins, 2017)

4.1.1 Inception Phase

The Inception Phase started already in the 6th semester by creating the project idea, followed by Project Description and the first version of Software Requirements Specification (SRS). Afterwards, in the beginning of 7th semester it was crucial to have the same understanding in the team about the idea and the project itself. Therefore, we have decided to make an Event Storming session when we generated different ideas for main project targets and features. The result of this session was the updated project description (Appendix A) and SRS (Appendix C) with clearly stated target groups – students and teachers. Further details about Event Storming can be found in the section 4.2.2.1 - Event Storming. Once we had a clear vision about the project and an updated list of requirements, we were able to prioritize features and set up the Product Backlog that can be found in Analysis section of Project Report.

Moreover, at the end of Inception Phase we agreed on SCRUM procedures (starting in the Elaboration phase), which are described in the section 4.2 - SCRUM. Considering the Time Plan in the Project Description, the Milestone #1 was successfully accomplished on September 16, so that we were ready for the next phase of Unified Processes called Elaboration.

4.1.2 Elaboration Phase

The elaboration phase started on 17th of September and the objective of this phase was to finish all must have functions from most important features:

- Account and profile management
    - register
    - update user roles
- Login / Logout
- Uploading video
    - share video
- Watching video
    - watch video
    - interact with the video

In the elaboration phase the group started with data modelling and architecture diagrams that were crucial part for implementing a system of this size. After this part, we moved on to domain modelling, creating use cases and we started to implement features. Every feature was sequentially analysed and designed in each sprint. With regular meetings and properly planned sprints we were able to finish all crucial parts for proof of concept till 30th of October. The proof of concept demonstrated a working Android application that can register a user who can watch and upload video.

However, we were planning to implement a cross-platform solution that can run on Android and also as a web application. The most crucial features support two platforms, but we found out that we won’t be able to continue to support two platforms due to lack of time. It turns out that even though Flutter can run one code on multiple-platforms it's still required to create alternative layouts that are very time demanding, especially for a web application and some features can’t be implemented in the same way for different platforms.

In case of the architecture, there was one change in the end of the elaboration phase where we decided to merge two services together (user service and group service), as we realized that both services are using same data and in this way we ensured that they won’t be interrupting each other during reading / writing processes.

4.1.3 Construction Phase

The construction phase started on the 30th of October and the objective of this phase was to finish all must have and should have features:

- Account and profile management
    - register *(was done in elaboration phase)*
    - update user roles *(was done in elaboration phase)*
- Login / Logout *(was done in the elaboration phase)*
- Managing video
    - change video attributes
- Searching for videos
    - search for video by search term
    - advance search
- Feedback on videos
    - rate video with like or dislike
- Groups
    - create groups
    - adding users to groups
    - setting instructor role
- Recommending videos
    - generate recommendations
    - personal preferences
- Uploading video
    - share video
- Watching video
    - watch the video *(was done in elaboration phase)*
    - interact with the video *(was done in elaboration phase)*

In the construction phase, the main focus was on recommendations, where deep analyses of the user and their preferences were needed to create an algorithm that is capable of finding the best matches based on the user's personal preferences and actions. We started by defining user actions and selecting the most important ones. Then it was needed to prioritize them and assign values to them. To improve the accuracy of the outcome, a time penalty was applied to each user action, to subtract the priority if the action is too old. To test this algorithm, we needed a lot of data about the user, therefore we decided to create a script that generates videos and users with their actions. As there was a need to focus on the recommendations feature more than we expected, we decided to reprioritize our backlog. We prioritize feedback over creation of groups as it has a bigger value for the recommendation engine. To finish the first release candidate, we moved the deadline from the 8th of December to the 13th of December and because of the lack of time we haven't implemented the groups feature, edit video and advance search feature.

4.1.4 Transition Phase

The transition phase was planned to start on the 8th of December but was moved to the 13th of December. The objective of this phase was to make the final acceptance tests and finish the documentation. The result of the tests is very positive, all must-have features have passed the tests and we decided to create an additional test tool that helped us to generate user data and videos for testing the recommendation algorithm. On the 15th of December, we had a meeting with the supervisor that helped us go through the final version of the project report and enhance some parts of it.

4.2 SCRUM

The Scrum framework was chosen for controlling the process of project development. There are other simpler alternatives of frameworks that could be used within this project, as the team consisted only of 3 members, like for example Kanban. Nevertheless, it was decided to use Scrum mostly due to the reason that the project was expected to be complicated which turned out to be true since the backlog has 9 features with 29 user stories in total, and also because all team members had experience with Scrum framework from work and previous projects. (Schwaber and Sutherland, 2020)

The details about how Scrum was used and adapted into this project are described in the following sections. Full documentation of meetings for each of the sprints can be found in Appendix D.

4.2.1 SCRUM roles

One of the principles that Scrum is based on is the division of roles - Product Owner, Scrum Master and Developers. For this project, Michaela was chosen as a Product Owner because of her experience with UX development, so that she was capable of understanding the needs of potential customers and representing them in the form of user stories. The Scrum Master role was assigned to Matej as he is considered to be a detailed-oriented person with a high demand for effective teamwork. Since, the Scrum framework is more suitable for teams with more members, Michaela and Matej were also a part of the Development team, where the lead was taken over by Michal as he is the most experienced developer from the whole team.

4.2.2 Working methodology

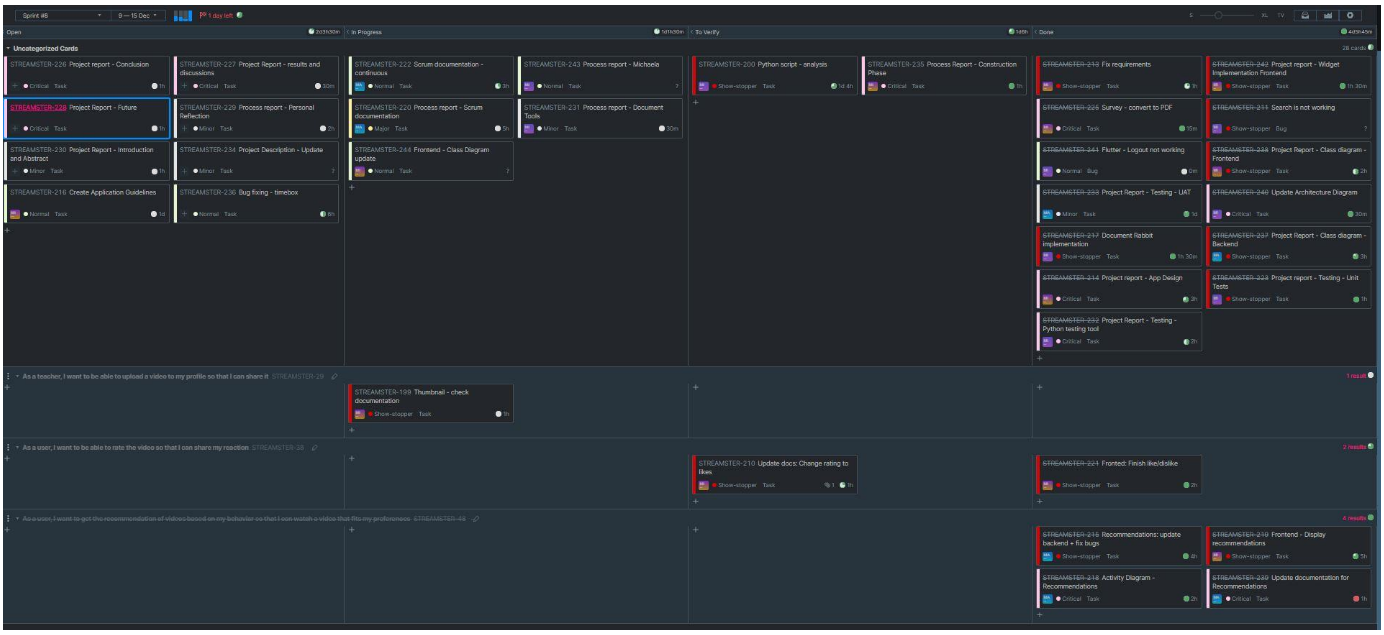
In case of working methodology, we were following the Scrum principles which are based on time-boxed iterations called sprints. The exact methodology was defined by the team at the end of Inception Phase and it is based on the following points:

- Sprints are 2 weeks long - starting and ending on Wednesdays
    - Out of 2 calendar weeks approximately 3 days are working days which equals to 24 working hours per person in one sprint.

- There is a Sprint planning meeting prior to each Sprint. Main goals of this meeting are the following:
    - To update the Product Backlog regarding the actual feature priorities which are then used to decide which feature is going to be selected for the upcoming Sprint
    - To create a Sprint Backlog with estimated tasks depending on team's availability in the upcoming Sprint
    - To set goals for upcoming Sprint which helps to evaluate the outcome of the sprint at the end
- There is a Sprint review meeting in the end of each Sprint which main goals are:
    - To review all tasks from Sprint Backlog - their status (not started, in progress, done etc.)
    - To check if the team manage to fulfil current Sprint goals
- There is one Retrospective meeting in each Sprint which purpose is:
    - To talk in team about previous Sprint - among others the typical questions during this meeting are as follows:
        - What went good in the finished sprint?
        - What could be improved for the next sprints?
        - Did we manage to improve our work in comparison with the previous sprints?
        - Was the task estimation and distribution appropriate?
- Each project day starts with Daily Stand-up when each member presents their progress on their current tasks in order to make clear understanding of current status within the team and if there are some obstacles the team is finding the solutions together.
- Total amount of tasks taken for new Sprint is adjusted to the actual team's availability
- Each task has its estimated amount of time to be needed to complete it
- Tasks are redistributed equally on each team member
- Tasks and progress are tracked by YouTrack end documented later in Excel sheets

Moreover, it is worth to mention that for the Project period it was decided to shorten the length of sprints to be 1 weeklong because of the higher amount of time reserved for this project in comparison with normal period of semester. That means that Sprint #6, #7 and #8 were only 1 weeklong, however the amount of working days in a sprint kept unchanged, which is 3 working days per person. By this change of length of last sprints, the team managed to estimate better the amount of tasks to be taken for the actual sprint.

To give an idea to a reader interested in using YouTrack, the following snippet from YouTrack Agile Board was taken during the last sprint. There is possible to track current status of the task, the assignee, estimated and spent time, type of task and task priority among other relevant information for having the process under control.


Sprint #8 | 9 — 15 Dec | 1 day left

Open 2d3h30m | In Progress 1d1h30m | To Verify 1d6h | Done 4d5h45m

Uncategorized Cards — 28 cards

Open:
STREAMSTER-226 Project report - Conclusion — Critical, Task, 1h
STREAMSTER-227 Project Report - results and discussions — Critical, Task, 30m
STREAMSTER-228 Project Report - Future — Critical, Task, 1h
STREAMSTER-229 Process report - Personal Reflection — Minor, Task, 2h
STREAMSTER-230 Project Report - Introduction and Abstract — Minor, Task, 1h
STREAMSTER-234 Project Description - Update — Minor, Task, ?
STREAMSTER-216 Create Application Guidelines — Normal, Task, 1d
STREAMSTER-236 Bug fixing - timebox — Normal, Task, 6h

In Progress:
STREAMSTER-222 Scrum documentation - continuous — Normal, Task, 3h
STREAMSTER-243 Process report - Michaela — Normal, Task, ?
STREAMSTER-220 Process report - Scrum documentation — Major, Task, 5h
STREAMSTER-231 Process report - Document Tools — Minor, Task, 30m
STREAMSTER-244 Frontend - Class Diagram update — Normal, Task, ?

To Verify:
STREAMSTER-200 Python script - analysis — Show-stopper, Task, 1d 4h
STREAMSTER-235 Process Report - Construction Phase — Critical, Task, 1h

Done:
~~STREAMSTER-213~~ Fix requirements — Show-stopper, Task, 1h
~~STREAMSTER-242~~ Project report - Widget Implementation Frontend — Show-stopper, Task, 1h 30m
~~STREAMSTER-225~~ Survey - convert to PDF — Critical, Task, 15m
~~STREAMSTER-211~~ Search is not working — Show-stopper, Bug, ?
~~STREAMSTER-241~~ Flutter - Logout not working — Normal, Bug, 0m
~~STREAMSTER-238~~ Project Report - Class diagram - Frontend — Show-stopper, Task, 2h
~~STREAMSTER-233~~ Project Report - Testing - UAT — Minor, Task, 1d
~~STREAMSTER-240~~ Update Architecture Diagram — Critical, Task, 30m
~~STREAMSTER-217~~ Document Rabbit implementation — Show-stopper, Task, 1h 30m
~~STREAMSTER-237~~ Project Report - Class diagram - Backend — Show-stopper, Task, 3h
~~STREAMSTER-214~~ Project report - App Design — Critical, Task, 3h
~~STREAMSTER-223~~ Project report - Testing - Unit Tests — Show-stopper, Task, 1h
~~STREAMSTER-232~~ Project Report - Testing - Python testing tool — Critical, Task, 2h

As a teacher, I want to be able to upload a video to my profile so that I can share it STREAMSTER-29 — 1 result
In Progress: STREAMSTER-199 Thumbnail - check documentation — Show-stopper, Task, 1h

As a user, I want to be able to rate the video so that I can share my reaction STREAMSTER-38 — 2 results
To Verify: STREAMSTER-210 Update docs: Change rating to likes — Show-stopper, Task, 1, 1h
Done: ~~STREAMSTER-221~~ Fronted: Finish like/dislike — Show-stopper, Task, 2h

~~As a user, I want to get the recommendation of videos based on my behavior so that I can watch a video that fits my preferences STREAMSTER-48~~ — 4 results
Done:
~~STREAMSTER-215~~ Recommendations: update backend + fix bugs — Show-stopper, Task, 4h
~~STREAMSTER-219~~ Frontend - Display recommendations — Show-stopper, Task, 5h
~~STREAMSTER-218~~ Activity Diagram - Recommendations — Critical, Task, 2h
~~STREAMSTER-239~~ Update documentation for Recommendations — Critical, Task, 1h

Table 2 - YouTrack agile board

4.2.2.1 Event Storming

Event storming is a method of discovering what events and interactions are happening in a certain domain. It was created by Alberto Brandolini to be used as a tool for discovering domain in Domain-driven-design. However, it can be also used as a tool for specifying software requirements or business process modelling.
It is a workshop-based approach and does not require preparation from the participants. The analysed process is "stormed out" as a series of domain events represented by orange sticky notes stuck to the wall. The sticky notes notation can be extended to describe users, commands, external systems or business processes in order to create an extensive and readable view of an analysed domain (Brandolini, 2020).
In our project, we have used Event Storming at the beginning of Inception phase to discover the processes and events in our domain - that is video streaming system. It especially helped us to align our vision of the system and discover different actions that users can do in our system. The result of the Event storming session can be seen on the picture below.

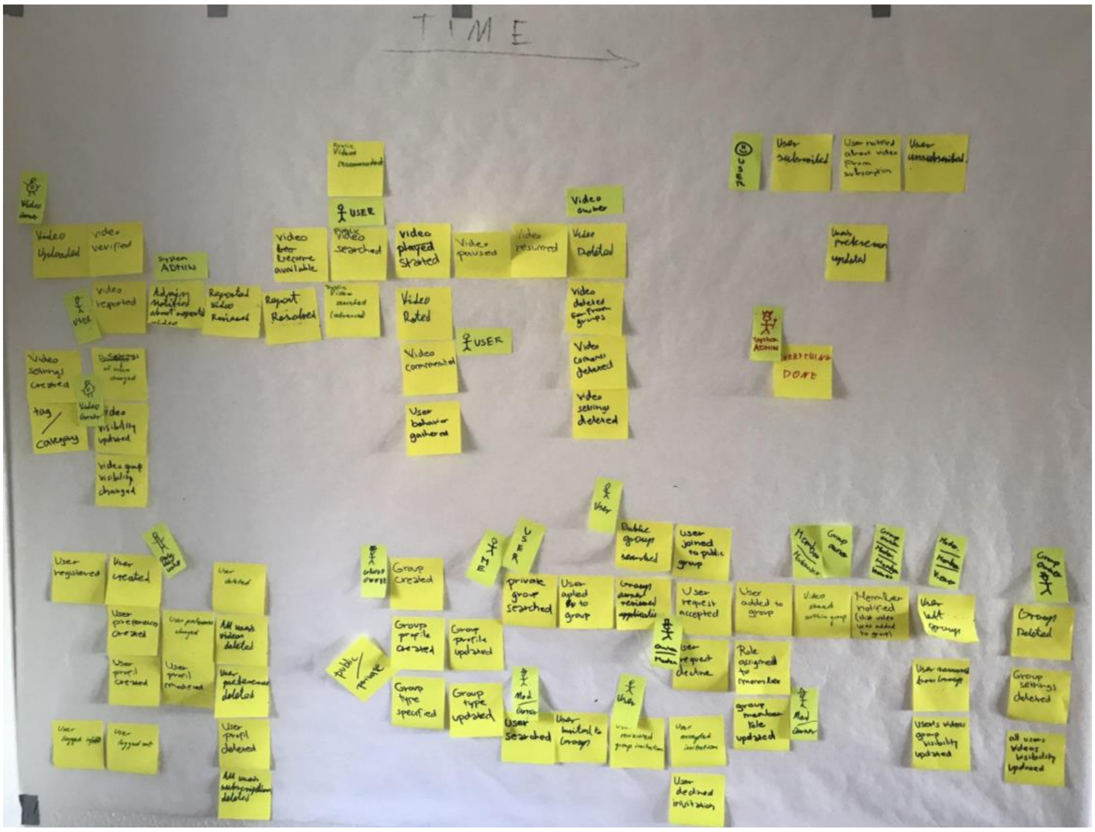
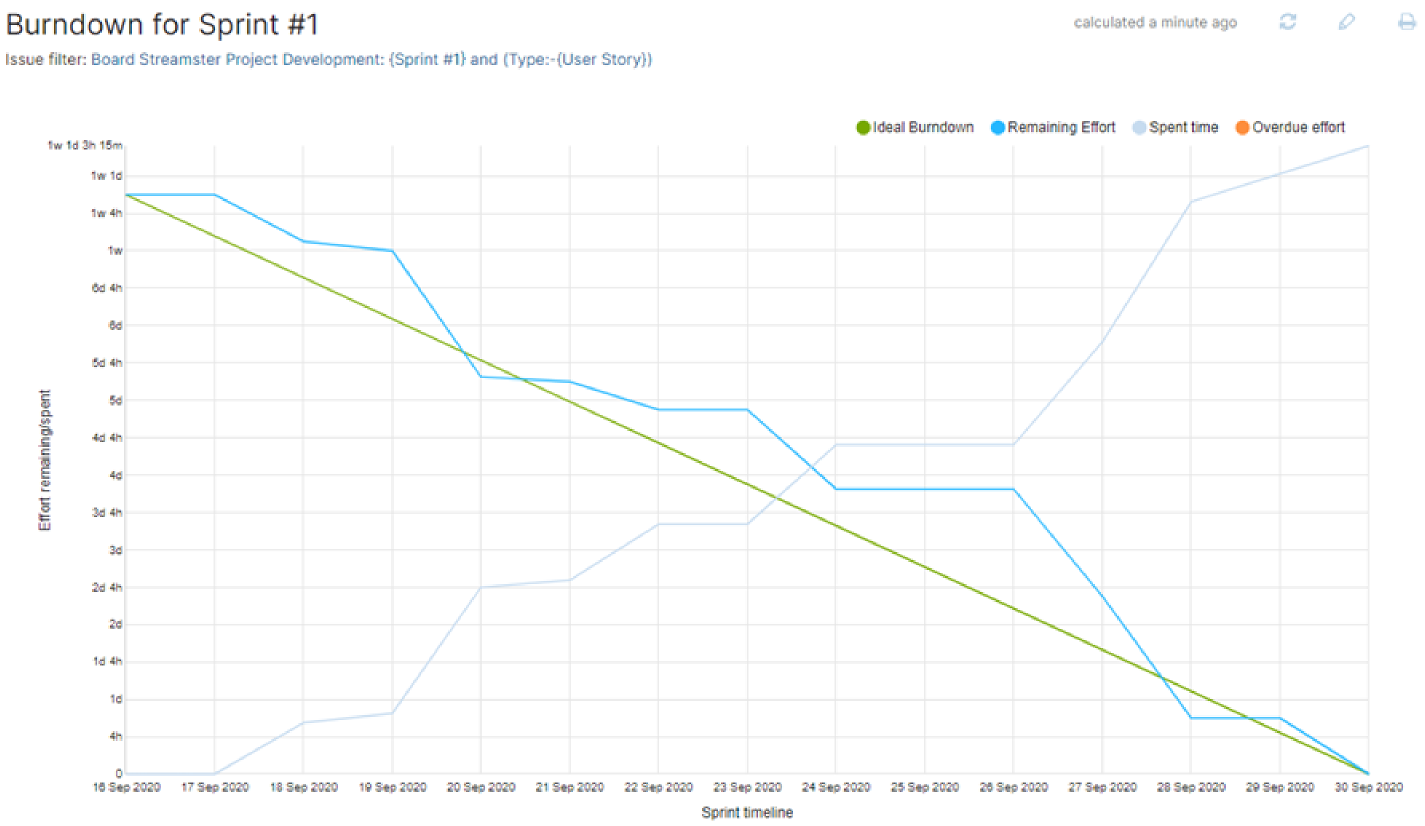


Table 3 - Event Storming

4.2.3 Burndown chart

In order to track the progress of the team in each of the sprints, the Burndown Chart was utilized. The following Burndown Chart is taken from the Sprint #1. Each of the burndown charts has remaining effort on the y-axis and the sprint timeline on the x-axis. The green line represents the ideal progress of the sprint as it distributes the amount of work equally to each of the days of sprint. The blue line reflects the remaining effort for each day, which can be understood as the actual progress of the team. Naturally, it is not as straight as the green line since the team was not working on this project every day in the week equally. Moreover, the line for Spent Time is shown in the chart. This line represents the actual team effort put into this sprint and often it is not coincident with the remaining effort because in some cases, tasks were not estimated accurately, and the team needed more time to complete the task. Therefore, as in the case of this sprint, the Spent Time line ends higher than the start of Ideal and Remaining Effort lines. In some of the sprints, it can be observed that the team failed to complete all tasks and there the Remaining Effort line does not end in the same point as the Ideal green line. In this case, unfinished tasks were completed in the next sprints.



Burndown for Sprint #1

Issue filter: Board Streamster Project Development: {Sprint #1} and {Type:-{User Story}}

calculated a minute ago

Table 4 - Burndown chart of Sprint #1

In conclusion, the Final Burndown Chart (16th September - 16th December) is provided below to give an overview of the overall progress during this project. In this overall chart, the number of remaining tasks is placed on y-axis. Therefore, this chart reflects the daily team effort committed for this project.

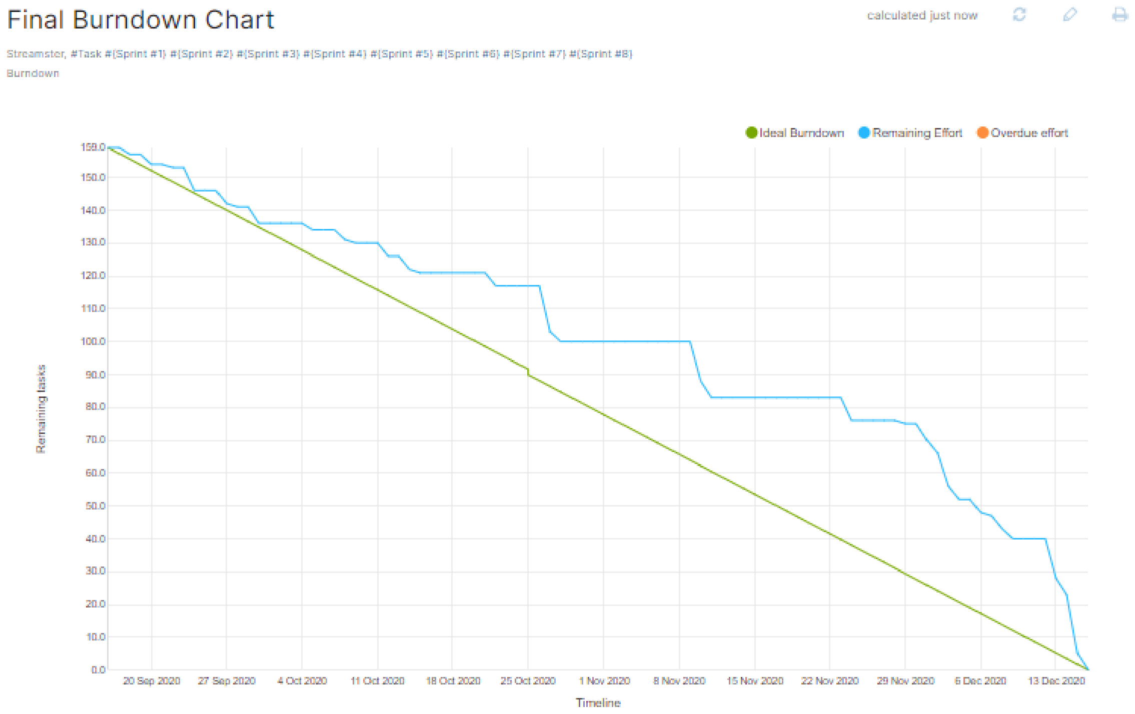


Table 5 – Burndown Chart Final

It can be observed that there are periods when the team was stuck with complicated features and therefore was not able to close any tasks. In particular, one of these periods was the time of designing and implementing the recommendations when the team had various difficulties to finish recommendations and had limited time because of other study activities accumulated at the end of the semester.

4.3 Technical tools

During project execution we have used a set of different tools to help us execute various tasks. To manage our work and Scrum we have used YouTrack. There we were tracking our sprints and status of tasks on the Agile boards. To create documentation, we were using Google Docs, where the possibility of simultaneous editing was very helpful. We have created all our analysis and design diagrams using Astah. Moreover, we kept our code base in a Git repository, hosted on GitHub. To write the code, we have used Intellij IDEA, as we have the most experience in using this IDE. During development we have also set up CI/CD pipeline using TeamCity as build server and Google Cloud Run as a deployment environment. To be able to deploy the services to the cloud, we have used Gradle, a build tool for Java to build our application from source code and JIB Gradle Plugin to create Docker images. To test our backend services, we have made use of Postman. To access our databases directly we used MongoDB Compass and Neo4j Browser. To communicate with each other when we couldn’t meet in person, we used Discord.

4.4 List of tasks and responsibilities

- Product owner (overseeing the product state, maintaining the product backlog) - Michaela
- Scrum master (scheduling, leading, documenting Scrum meetings) - Matej
- Frontend - Michaela, Michal
- Backend - Matej, Michal
- Persistence - Michal, Matej
- Recommendations - Matej, Michaela
- Testing tool - Michaela, Michal

5 Personal reflections

5.1 Michaela Golhova

This project was the most challenging one, as we needed to learn additional technologies and languages to accomplish our goal. We set a lot of requirements and we had too high demands for ourselves that pushed us to work harder to finish as many features as we could during a too short period of time for a project of this size. I am very satisfied with how much I learn about a cross-platform solution with Flutter. This project allows us to go deeper to advance features in this technology as was for example bloc pattern for state management, libraries for reading and displaying video or image files, complex widgets, or alternative layouts for responsive applications. Unfortunately, we abandoned some features as support for a web application or some other functionalities due to lack of time, but yet we managed to handle and learn many other technologies like Neo4j, MongoDB, Python, RabbitMQ, Spring or work methodology like Event Storming. Our team was very cooperative, and planning went very well because we knew each other and we were working together since the first semester that definitely brought us solid results for this project.

5.2 Michal Pompa

I think that this project was a good conclusion of my studies at VIA University in Horsens. We have built the platform using the knowledge we've acquired during previous semesters, but the most importantly we have built on that knowledge to create a complex system that is using a wide range of the technologies and tools that are often used in real life application in similar form. In my opinion, the fact that we have been in the same group since the beginning of the studies was very beneficial. We understood each other perfectly as we were already used to working together. Thanks to that, our work progressed very rapidly, even though we have learnt a lot of things along the way. I hope that I will be given an opportunity to make other projects in this team in the future.

I am also very glad that we have developed not only our technical skills, but also soft skills. I have found the Event Storming session very instructive and I would like to use this technique in my future work.

To conclude, I am quite surprised how much we have managed to accomplish in just a few months given the very ambitious scope of the project and time limitations. I am very satisfied with the results and happy that I could be part of such a great team.

5.3 Matej Michalek

In my opinion, our Bachelor Project managed to test us from all possible perspectives. Due to a huge variety of defined requirements, this project was challenging us from the first and until the last day of the project, so we got a valuable opportunity to acquire knowledge and experience in a wide range of software topics.

Personally, I appreciated working on micro-services architecture with asynchronous communication between them using message queues, two different database paradigms - document based MongoDB and Neo4j graph database and working on the topics about persisting large files and streaming videos.

Besides that, I feel that I have strengthened my skills of working in an agile team. I had to look at the process also from another perspective - perspective of a Scrum Master who always tried to get the best from the team. As a Scrum Master, I was satisfied with our work and I believe if we had had more time to implement it, we would have been able to complete it finally and possibly even to launch it onto the market.

In conclusion, I enjoyed working with Michaela and Michal and I hope we will have another chance to work on some project together in the near future.

6 Conclusion

The final version of this project provides a usable application where teachers can upload videos that students can watch, search or get them recommended. That makes our goal to be completed and it showed us that we can solve and execute complex problems in programming. Therefore, the project can be concluded with a positive result. Since we knew each other from our first days at VIA, it was not that hard to estimate how much and how fast we will be capable of doing. Nevertheless, the implementation of web application alongside with Android app was over our capacity, when we understood that it would be an overkill for this project in a team of 3 students. The overall process went without any serious difficulties and was adjusted throughout the project. We have managed to successfully accomplish all planned milestones. Only in the case of 1st Release Candidate we got delayed due to various semester-end duties from other subjects. In conclusion, it was demonstrated that the team managed to get the project done successfully and all team members have improved their skills and have acquired new knowledge valuable for their further studies and future work opportunities.

7 References

Brandolini, A., 2020. *Introducing EventStorming*. Available at: <https://leanpub.com/introducing_eventstorming/read_sample>.

Osis, J. and Donins, U., 2017. Chapter 2 - Software Designing With Unified Modeling Language Driven Approaches. In: J. Osis and U.B.T.-T.U.M.L.M. Donins, eds. *Computer Science Reviews and Trends*. [online] Boston: Elsevier.pp.53–82. Available at: <http://www.sciencedirect.com/science/article/pii/B9780128054765000022>.

Schwaber, K. and Sutherland, J., 2020. *What is Scrum ?* Available at: <https://www.scrum.org/resources/what-is-scrum>.

8 Appendices

Appendix A: Group Contract – GroupContract.pdf

Appendix B: Project Description – ProjectDescription.pdf

Appendix C: Software Requirements Specification – SRS.pdf

Appendix D: Scrum documentation – Scrum.zip