Computer Science

System Development Report

3rd Semester

Alexander Zahariev
Arnas Sokolovas
Jindrich Jehlicka
Marius Constantin Untaru

The following report of System Development consists of two parts. First part of the report will cover and focus on the theoretical part of the subjects and aspects covered in third semester.
Second part of the report will cover the process of the group work and arguments on the choice of methods group used with a following conclusion.

Contents

Chapter 1

Idea Generation

At the beginning of every group project that is aiming to develop a software product, the team is looking for a mission, a common vision. As everybody has their own ideas about what could be developed, what could be useful for their own or other people's needs.

The idea generation process is made up of multiple steps. Starting with capturing customer's needs and going all the way up to refining the final idea.

1.1 Brainstorm

One of the major tools that most groups use as a first step to generate ideas is brainstorming. While brainstorming, people should have no boundaries, no hardware perspective, no time perspective or anything else. The more ideas, the better as afterwards comes the categorizing phase. Here the team categorizes the ideas and then explores them by imagining scenarios where the ideas would be put in practice. Every scenario has a persona which is used to represent a specific type of person, for example, a student. To make scenarios understandable for the whole team, storyboards could be created for the selected ideas. Storyboards in general visualize the flow of acts in the specific scenarions within the target group or personas. Several of more sequential pictures can help the development team to understand and discuss the idea more thoroughly. As a final phase, the ideas are refined by evaluating and putting them in a business context and developing them to a technical prototype stage.

1.2 Business Model Canvas

Another great tool that helps with making a strategy for the chosen idea is business model canvas. It is composed of key components that are necessary in every business and it helps to define an implementation strategy for the idea. As shown in the Figure 1.2.(see p.5), there is a strong connection between all the components. From specifying how the company's resources are to be used in relation to the company's partners in order to achieve the production activities and create value for the customers, to how the final product is delivered through different channels while maintaining good customer relationships.

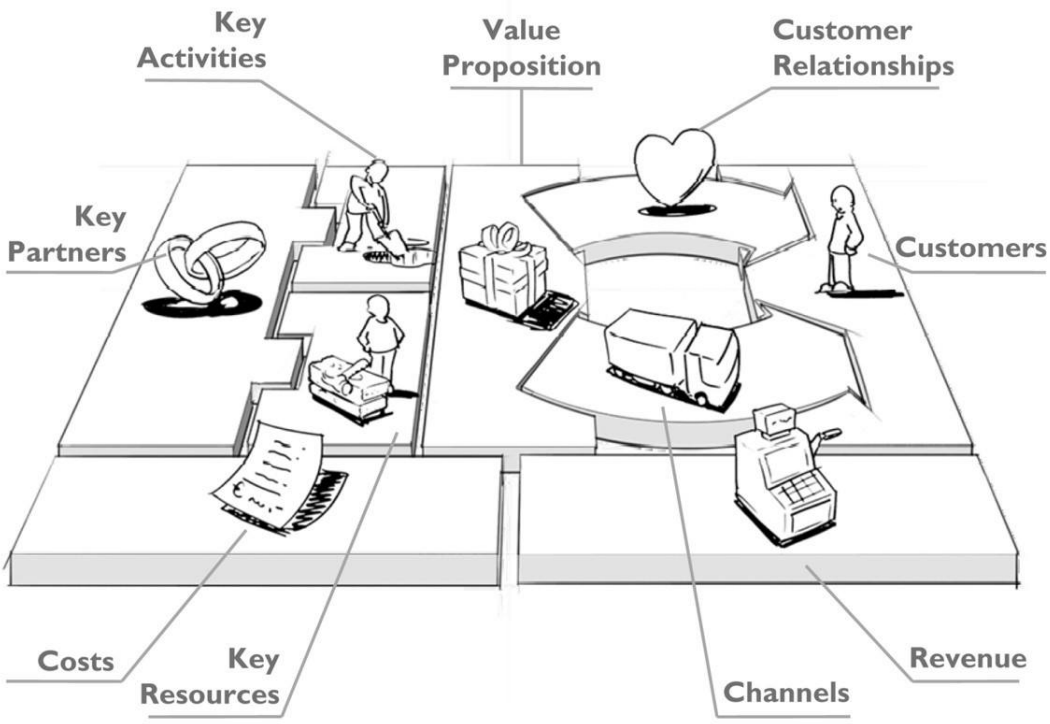

Figure 1.2: Business Movel Canvas

1.3 Prototyping

Prototyping is an important element in the idea generation process. It can be used many times during the development phase of the project to help creating a better vision of the final product. Based on the scenario, the ideas can now be put in a design and technology perspective. The ways to do that are Horizontal and Vertical prototypes.

Horizontal prototyping is used to combat design challenges and refine the prototype in accordance to its fidelity, which reflects how close is the prototype to the final design. The best techniques are paper-prototyping/storyboards, static images on the device and Interactive prototypes like video or Microsoft PowerPoint presentations. All of them have different fidelity levels and are used to show functionality, usability and visual aspects.

To get the most from the prototypes it is essential to evaluate them and start by creating evaluation criteria. As an example, who is going to evaluate them (Users/Experts) and which techniques are to be used.

One of the techniques is Guerrilla testing. It is a good technique to find usability issues and solve them fast as Guerrilla testing is very easy to do and can be done by anyone. It is only necessary to give a couple of tasks to the users, observe their interactions and ask about the experience. Another similar technique is Think Aloud Test where the main difference is that the test participants are asked to express their thoughts verbally while using the application.

Chapter 2

Process Models

There are different process models that can be used throughout the software development process. The models can vary in the corresponding approach that the development is organized and planned. Various activities such as requirements specification, design of the product, implementation, testing and others, are involved in most of the projects and it is necessary for a decent and legitimate product development.

There are few most popular process models that are widely used in software development section. They can be described as basic, such as Waterfall, Evolutionary Development, Component Based Development, or modern, more advanced models like Unified Process(UP), Agile(eXtreme programming and Scrum). Each of the models has its own advantages and disadvantages and it is up to the project team to decide which approach to choose. Following section will provide brief descriptions of the models with the focus laid on the most popular ones.

2.1 Basic Process Models

Basic models such as Evolutionary Development and Component Based Development are straightforward. Evolutionary Development, as its name implies, says that the whole process is covered by iterative phases of the product development on the customer's feedback. Testing of the product is an essential part of the Evolutionary Development due to its incremental development.

Another basic model is called Waterfall (see Figure 2.1 below). This model has a linear approach and it is structured into numerous phases. Gathering all the requirements and documents, making a design, implementation and system testing, user acceptance testing, bugs fixing and in the end delivering the product to the customer. Each phase has to be finished before the next one can start. This model has a couple drawbacks. All the changes are very difficult because there is only one pass. It is also very expensive to fix all the errors that are discovered late.

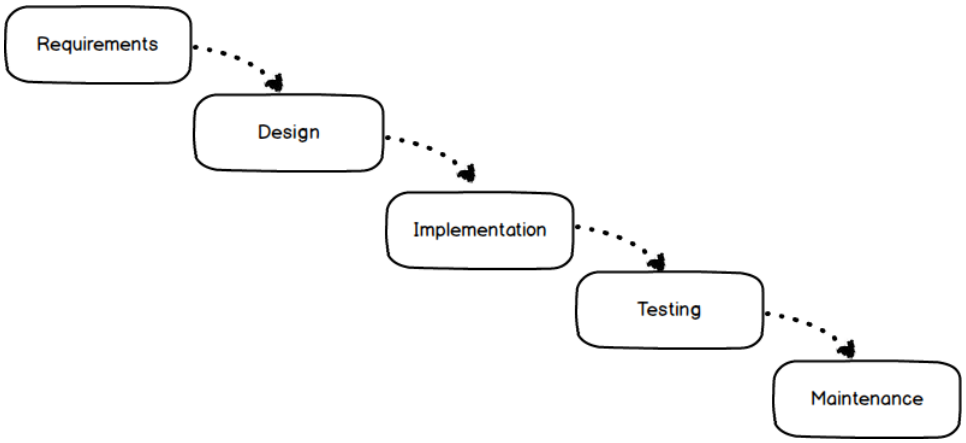

Figure 2.1: Waterfall Model

Most of the modern models can be described as a combination or mixture of some of the basic models mentioned before. While choosing to work with any of the modern models, it is important that all of the advantages and disadvantages are thoroughly studied and understood. Following section will describe some of the modern models, such as Unified Process and Agile.

2.2 Modern Process Models

Unified Process(UP) is an architecture-centric and use-case driven model. UP is divided into four phases. This software evolution methodology lays high amount of focus on documentation and highly involves client which helps to remove risks from the project. Nonetheless, it requires high degree of software development and sometimes might cause troubles understanding the methodology itself. Following, four phases of UP will be given and briefly described.

- ***Inception*** - scope of the system and business case are defined.
- ***Elaboration*** - major part of requirements are gathered. Use-case diagrams are created.
- ***Construction*** - iteration, implementations based on the foundation of Elaboration phase.
- ***Transition*** - system deployment, feedback gathering and changes of the requirements if needed.

Agile methodology is also known as an innovative software development approach due to its short time software creating “boxes”, also known as iterations. Iterations let the development team to deliver small or bigger parts of software implementations of time boxes such as one week or bigger such as up to a month.

It is highly adaptive software development process which means that responds to changes is not an issue. While working under agile methodology, it helps and builds team’s communication and collaboration skills because of the constant team involvement in the progress of discovery and teamwork.

Challenges met while building software under agile methodology are such as, requirements that the team members have to be highly or skilled enough, as well cross skilled for the chosen Agile framework. As well the quality of the documentation might lack efficiency due to extensive work on the software part.

Following section of the report will present most widely-used Agile methodology framework - Scrum framework.

2.3 Scrum

Scrum is a great tool and a framework for complex software product development. It is simple to understand framework, but difficult to master. The whole process of Scrum is a flexible framework which is able to employ different techniques and processes while building a product. Whole framework can be also defined by looking at its three main sections with corresponding sub-sections:

- ***Scrum Team***
  - The Product Owner
  - The Development Team
  - The Scrum Master
- ***Scrum Events***
  - Sprint
  - Daily Scrum
- ***Scrum Artifacts***
  - Product Backlog
  - Spring Backlog

Following, brief and adequate explanations of some of the most complex parts and sections of the Scrum will be given.

All Scrum teams are self organised and cross-functional, meaning that different sets of people can come up with different ideas while coming from different areas of an organisation with different sets of skills. Teams as such are tend to place more emphasis on communication and work together to accomplish goals. Most of the Scrum Team roles speak for themselves giving a self-definition such as Scrum Master. It is a role where one is responsible for ensuring Scrum is understood. Scrum master does it by making sure that the Scrum team is following rules and practices.

Different events occur during the framework of Scrum. Most important and inevitable event in Scrum is Sprint. There is more than one Sprint, each different and responsible for different things, new iterations, planning, reviews and similar. It is important not to set Sprint goals that exceed huge amount of time and that the goals of the Sprint are not changed.
Daily Scrums is another important event which occurs daily within the Development team where the the team synchronizes their work and plans the following tasks for upcoming 24 hour period.

Scrum artifacts provide key information for Scrum Team about the product which is under development. It consists the information about activities which has to be done or planned.

Product Backlog is a list of features, functions, requirements that are requested to be seen in the end product. The Product owner has the responsibility to take care of the Product Backlog. It is also described as a “living” artifact while the requirements for the future product always change and vary. It changes due to the need of constantly identifying if the product is competitive enough, useful or applicable.

Sprint Backlog is quite different from Product Backlog, while it is a backlog of set of items from Product Backlog, selected for a certain sprint. It is also defined as an estimate for the development team about the future functionality of the product.

A powerful tool that helps the team to stay on track is the Burndown Chart. It shows the amount of work that has to be done in relation to an amount of time that fits the project schedule. This way every team member is aware of how the situation the team is in.

Chapter 3

Planning and Quality Assurance

3.1 Planning Poker

Planning Poker is a tool which can be used during software development to estimate time of every task. Each player gets a similar card deck where every card has a specific value. Most common values are 0, 1, 2, 3, 8, 20, 40, 100. These values represent a unit, either minutes, hours,days or story points. It depends on the team and which unit they estimate.

All the team members discussed features that should be included in the program and asked questions about it. After discussion, the game takes place and all the features that were discussed are estimated one by one. The players estimate how long it will take to implement given feature by one of their cards. Every player secretly picks a card and reveal it at the same time as the others. If every played card has the same value, it becomes the estimate. If the played cards have different values, player with lowest and highest estimates discuss their reasons for choosing their number. After that the estimating repeats until all the played cards have the same value. The following image below represents the basic card deck of Planning Poker game.

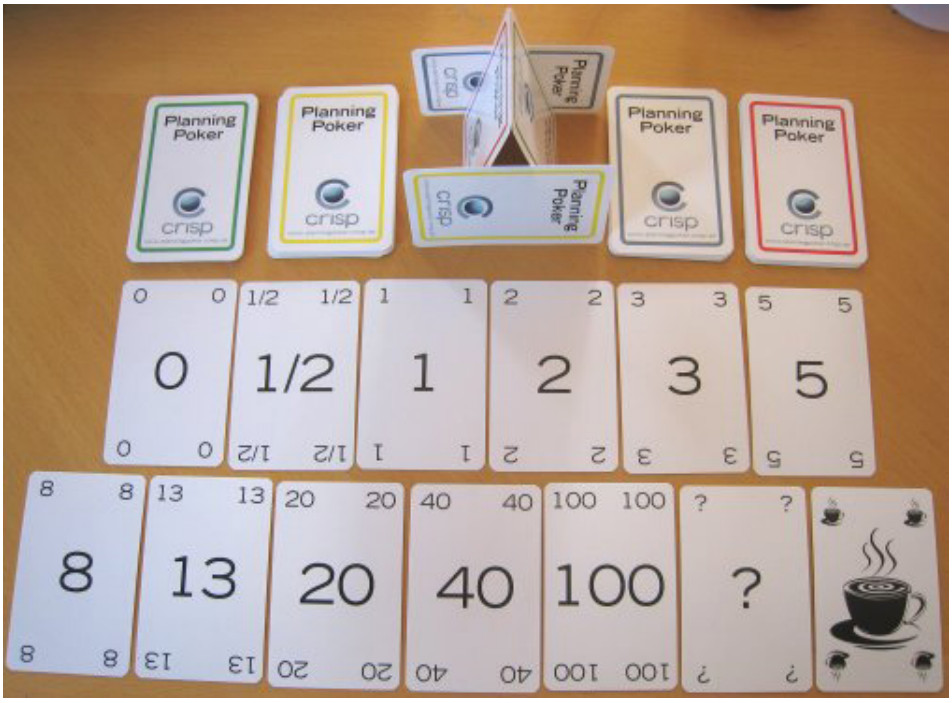

Image 3.1: Planning Poker
<https://upload.wikimedia.org/wikipedia/commons/e/eb/CrispPlanningPokerDeck.jpg>

3.2 Burndown Chart

Burndown chart is great tool to visualize current and ideal progress of the project. The group can use it to find out if the ideal burndown is followed.

The basic principle of the Burndown chart is to have two lines, one that shows the ideal progress and second that that presents the actual progress. In an example below (see figure 3.2), the burndown chart for Project X Iteration 1 is made. Reading from left toright, indication of start and end of the iteration can be found. Both of them are connected with two lines with markers. The perfect results can be achieved once the ideal line with markers is perfectly alligning with the actual line of progress.

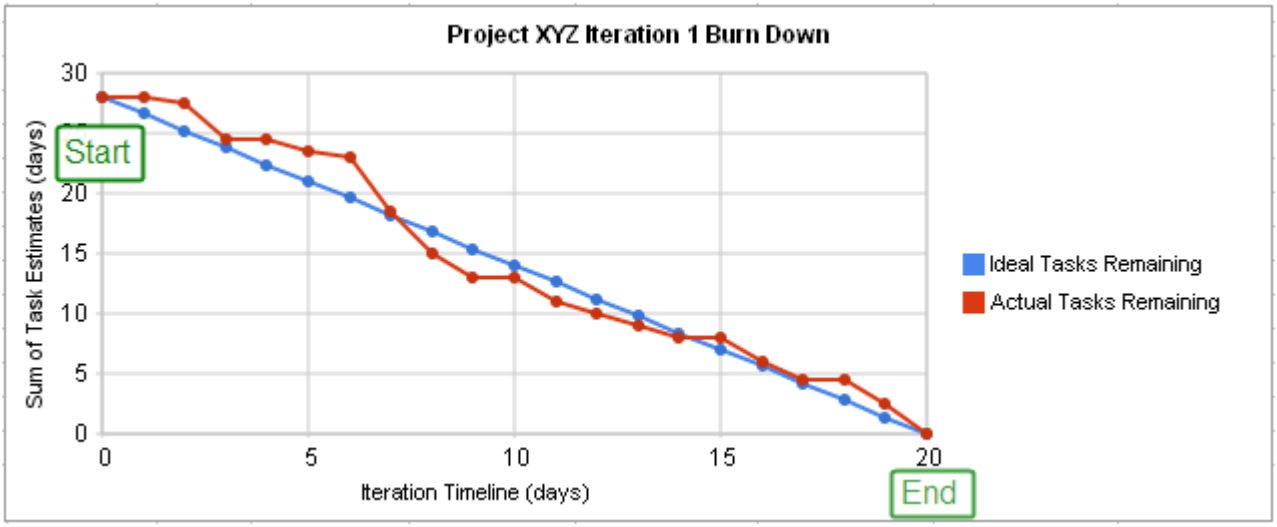

Figure 3.2: Burndown Chart

< https://upload.wikimedia.org/wikipedia/commons/8/8c/Burn_down_chart.png>

3.3 Quality Assurance

Quality assurance(QA) can be defined as the activities or processes which helps to guarantee that the desired products or services will meet all the quality requirements.

It is a process which is focused on the development of the product. Quality assurance is great tool to save time and money because the cost of finding and fixing problems and errors increases over the time. Prevention is cheaper than repair.

Quality assurance can be often confused with quality control. The main aspect that differentiates quality assurance from quality control is that quality assurance is done during the project to ensure the product will meet all the quality standards that are required. However quality control checks if the product met all the quality standards after the product is finished.

Chapter 4

Quality Criteria and Architecture

4.1 Quality Requirements

It is important whenever a customer is satisfied by the product and the product meets customer's requirements. Quality criteria, or quality requirements define all the characteristics and qualities the product must reach in order to satisfy customer's needs. Requirements may differ both in range and complexibility. One might be defined as a simple and abstract statement, while the other might be stated as a complex mathematical function. As well they might be categorized into functional and nonfunctional requirements.

Functional requirements straightforwardly state what your system should do or specify a specific function. To have a better understanding, let's have a real life example of a functional requirement. A yogurt carton has a simple requirement - "ability to contain yogurt inside the carton without leaking". As an another example of something more software based, let's say we have a shopping system. One of the functional requirements would be "add customer", or "print invoice" etc.

Nonfunctional requirements cover the rest of the requirements which are not covered by functional requirements. The purpose of nonfunctional requirement is to specify the criteria that can judge the system and have nothing to do with behaviour of the system like functional requirements do. For example, the shopping system before had a functional requirement for "add customer". For this case, the nonfunctional requirement for the system would be, for example, "data received in the database should be updated within 3 second for all the users who are accessing it". Nonfunctional requirements can cover lots of other typical areas of systems such as security, maintainability, usability and others.

As we talked before, in order for customer to be happy by the system or a product, it must meet the requirements. Once the system is implemented, the developers must prove the functionality of the system can satisfy the customer. Acceptance criteria is a great tool for proving how each criteria of the product owner is satisfied. It as well helps the developer team to see how a specific function or a feature will work from user's perspective. As an example for the shopping system and a function of adding a customer, acceptance criteria could include a user not being able to add a customer without completing mandatory fields. Or as an another example, payments can be made via PayPal etc. Acceptance tests as well form tests itself by confirming that a specific feature or piece of functionality is working and complete.

4.2 Architecture Pattern

Decision of which architecture pattern to use while building software system, can influence the price or quality of the product. It can also help while splitting work between developers and help while solving issues and problems in general. Following, three architecture patterns covered in curriculum will be briefly described.

MVC - is a pattern that separates the application into three components:

Model - data and behaviour, independent of the UI.
View - the display to the user.
Controller - responsible for handling requests.

Each of the components are handling certain aspect of the application. MVC architectural pattern is also one of the most frequently used as a web development framework.

Layered Architecture - focus laid on grouping functionality within the application into definite layers which are stacked vertically on top of each other.

Client-Server - it is a type of architecture where one program (client), requests a service or resource from another program (server). The main principle of client-server architecture is, that once the connection is established by the client program and server has fulfilled the request of the client, the connection is terminated. This type of architecture has become very popular in network computing.

Chapter 5

Reflection on Choice of Methods

The following chapter will cover most of the methodology used throuout the project. Reflection of the process will be given within corresponding argumentation on choice of methods with a following graphical presentations where needed.

5.1 Brainstorming

At the very beginning of the project an idea generation process was necessary, in this case, to establish a common goal for the whole team where the System Development, Programming and Technology concepts would be applied.

Starting with a brainstorm in which every team member noted down his own ideas about what software product should be created, and followed by the ideas being shared, discussed and put in categories. The ideas were limitless as some of them were intangible, but this was a tremendous help that kept the aim high and find the best possible idea. On the other hand, some were tangible and somewhere in between the two, the uncertainty depending on different factors, for example, how much time would take to accomplish it.

To help decide which ideas should be followed, scenarios were imagined for when these ideas would help people with their needs. They were concluded under the form of storyboards which implied using personas.

The persona that was created for the final idea can be seen in the Appendix A 1.1. John is a persona and he performs or can be seen as a potential user of the system. Personal information, personal interest and lifestyle were described along with the area of concern where the final idea would fit John's needs.

In the figure 5.1 p.14, an example of a storyboard can be seen, presenting the flow of acts in the scenario that the persona, John, is doing. Briefly explained, the first two scenes show John as he is moving in a new apartment and he wants to get rid of the old furniture. He comes up with the idea of selling them on the internet with the use of the "Auction System" system.

The other two scenes present him as he posts the sale on the system along with a picture and a preferable price. Then he agrees to sell it to the highest bidder of 250 kr.

From only four sketched scenes and four lines of text, this idea's concept and highlights can be visualised, making it very easy for the whole team to understand.

Figure 5.1: Storyboard

5.2 Playing the Planning Poker

Planning poker has been chosen in our project to estimate the time of each task and upcoming sprints. It was decided to play Planning Poker on daily group meetings, whenever some decision had to be done. The game was played throughout the project sprints which are described in the following sections of the report. The results of the game helped the group to make a better time plan and schedule for all the tasks. Although not all tasks followed the estimation time made during the game, the group thinks the game is a powerful tool which can help solve time issues. The following image 5.2. p.15 presents the demonstration of the group playing Planning Poker for a specific task.

Figure 5.2: Playing planning poker

5.3 Scrum Task Board

Once the final idea was visualised and thoroughly discussed and analysed, the SCRUM Task board (see. image 5.3 p.15) was made. It contains of the tasks used during sprints, while group members continuosly change and update it, based on estimations and changes made, or new tasks created. It is a great tool because it is simple and easy to use, letting the whole team be updated and familiar with the existing tasks, updates and changes.

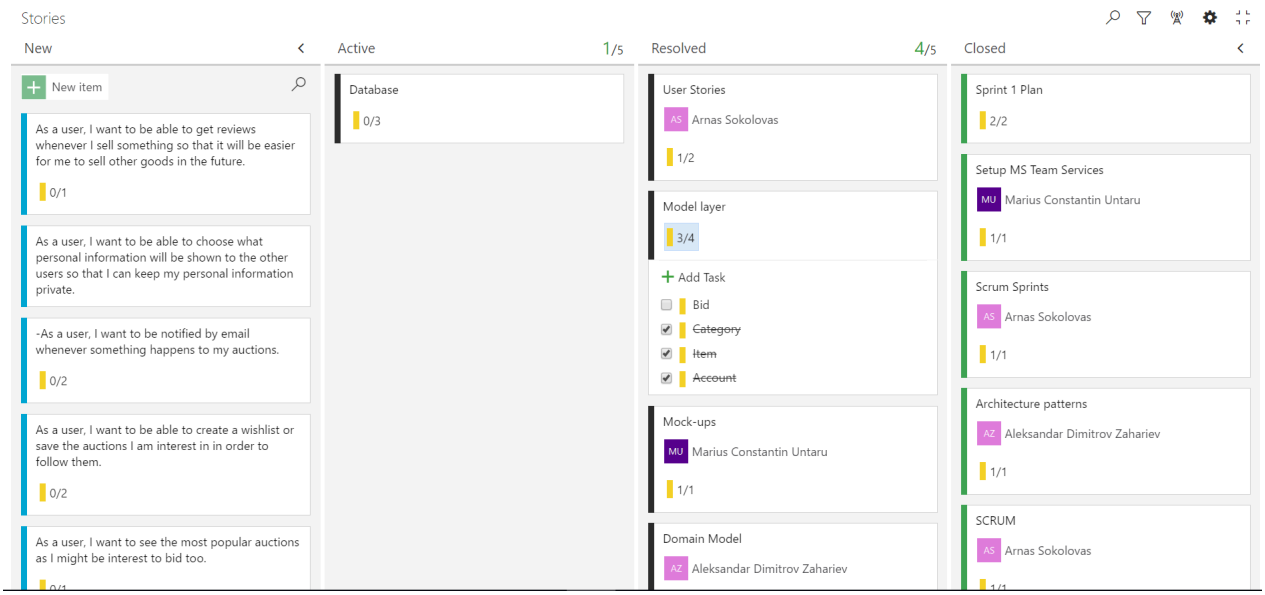

Figure 5.3: Screenshot of Scrum Task Board

5.4 Mock-up

Part of the Horizontal Prototyping, the selected final idea was further developed and visualized by creating a mock-up. Prototyping mock-ups is a great tool which helps to have a foresight of how the final product could look like and to present some functionality and usability aspects.

As it can be seen in the figure 5.5 below, the home page of the system includes different functionalities:

- Search bar - to help users find a specific product;
- Log in button - that implies multiple accounts for multiple users;
- Categories - for easier products browsing;
- Popular auctions - to show products that the users might be interested in and enable them to bid directly.

This is how the final version of the product was imagined (see figure 5.4 below) to look like at the end. It helped with guiding the group through the system production and defining the sprint plans.

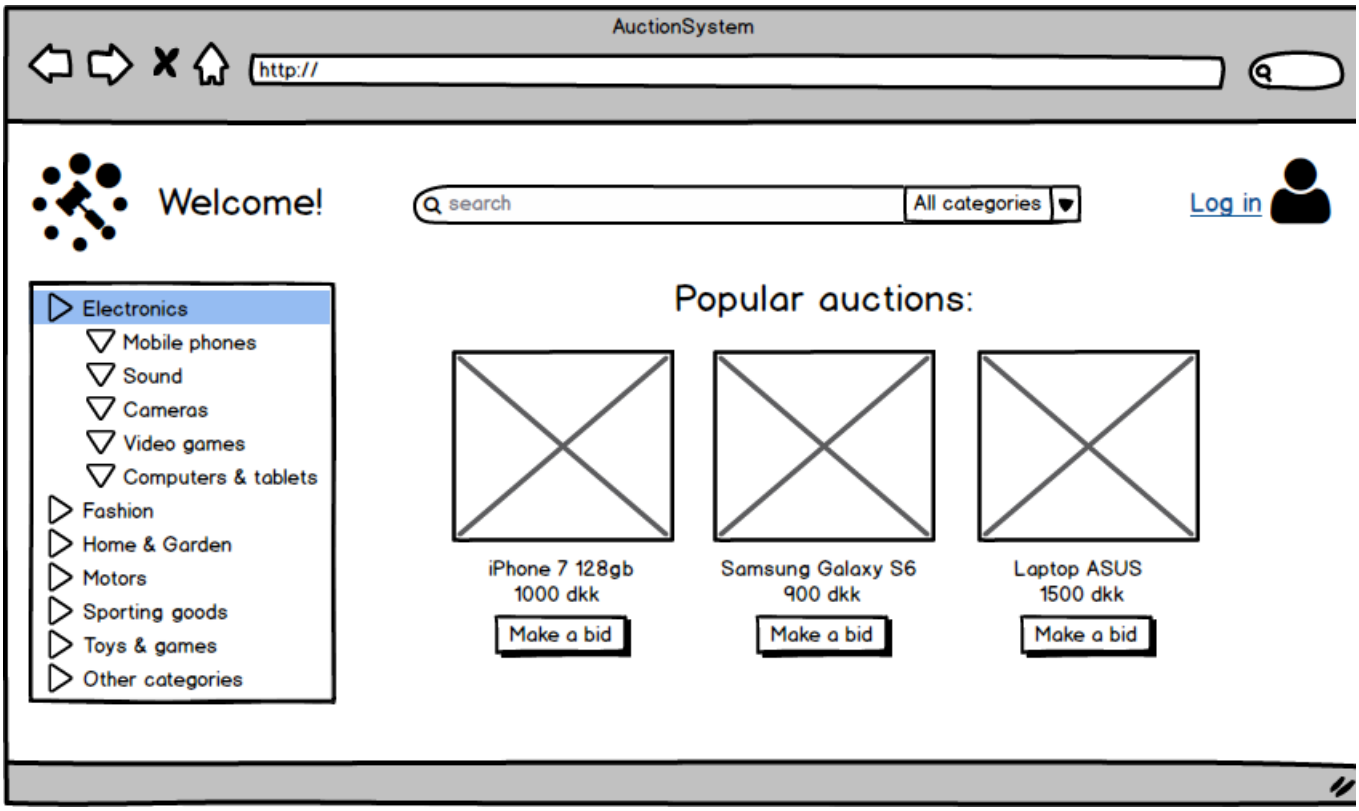

Figure 5.4: Mock-up of the Auction System

5.5 Sprint 0

To get a good head start and to get the whole team familiar with the Scrum procedure, it was decided to include a Sprint zero. At this stage, the team needed some defined rules that had to be respected by every member, also the tools that were needed to build the project were set up and a bit of planning was made too. The Group Contract (see Appendix A 1.2) was the key element in defining the team. Rules about what a member of this team should do or not to do, were specified and agreed upon by signing the contract. This helped to boost the confidence and motivation of the members.

The tools that were necessary to develop the project were downloaded and installed, for example, Visual Studio was the tool that was used to develop the program in C#, also Visual Studio Team Services was used to help the team with organizing the project with burndown charts, task boards and a product backlog. Github was another tool that allowed the team to have a common main branch of the program and create other small branches where features would be added to the program by eventually merging them to the main branch. Also making the code visible and available for everybody from anywhere which gives a great benefit to the project's progression.

The Product backlog was built upon the gathered requirements and the user stories that were created by the team, since a product owner was absent in this case. As they can be seen in the Appendix A 1.3 , the user stories, were made with the help of a simple template:

*As a *user type*, I want to *perform a certain action* in order to *achieve a certain goal*.*

The benefit of using this template was that it made it easy to think about small, but significant details of the program and create a list with a decent amount of user stories that would not leave room for uncertainties.

5.6 Sprint 1

Right after Sprint zero was finished, group meeting took place to share and discuss the process and tasks that were made during initial sprint. Following, the meeting continued and the discussion about Sprint 1 started. Once again it was essential to discuss most of the posibilities which would help realise the project idea. Tasks were assembled and placed on the Scrum Board in order to keep track on current or future tasks of the sprint. Few options of how to proceed with the software implementation were taken into consideration and the final decision was to implement the server using Web API technology. The main reason of the decision to use Web API was due to the previous experience working with and knowing that it is a newer technology than WCF. The following presents several sprint goals:

- Creating Web Api project;
- Create models layer;
- Create database;
- Implement controllers;
- Testing and debugging;
- Refactoring;

After the project was created, identity models were extracted into seperate projects. Following, the database migrations were setup and model, controller layers were created, following with the testing and refactoring of the project.

5.7 Sprint 2

At the beginning of the second sprint the group meeting took place to discuss tasks for this sprint. The goal of this sprint was to create web and windows applications. Planning poker took place again in order to make estimates for upcoming tasks. Based on the game results the time schedule was made and updated. Due to the difficulty of some tasks, this sprint was estimated to be the longest one. It was decided that daily group meetings were necessary to be done to fully understand, fulfill and update the tasks. The whole process of sprint two was quite successful although the time schedule was not perfectly followed, therefore there is a big difference between actual and ideal time left in burndown chart. Following, some of the sprint goals and tasks are presented:

- Implementation of Authentication;
- Implementation of Routes;
- Implementation of Login and Registering;

The following figure 5.7 below will present the ideal and the actual burndown charts. As it is shown, there lines with markers do no allign, meaning there ideal burndown was not a match for an actual, saying that the time spent on tasks did not match the estimated time during the planning poker session. The chart was build based on the template form the lecture.

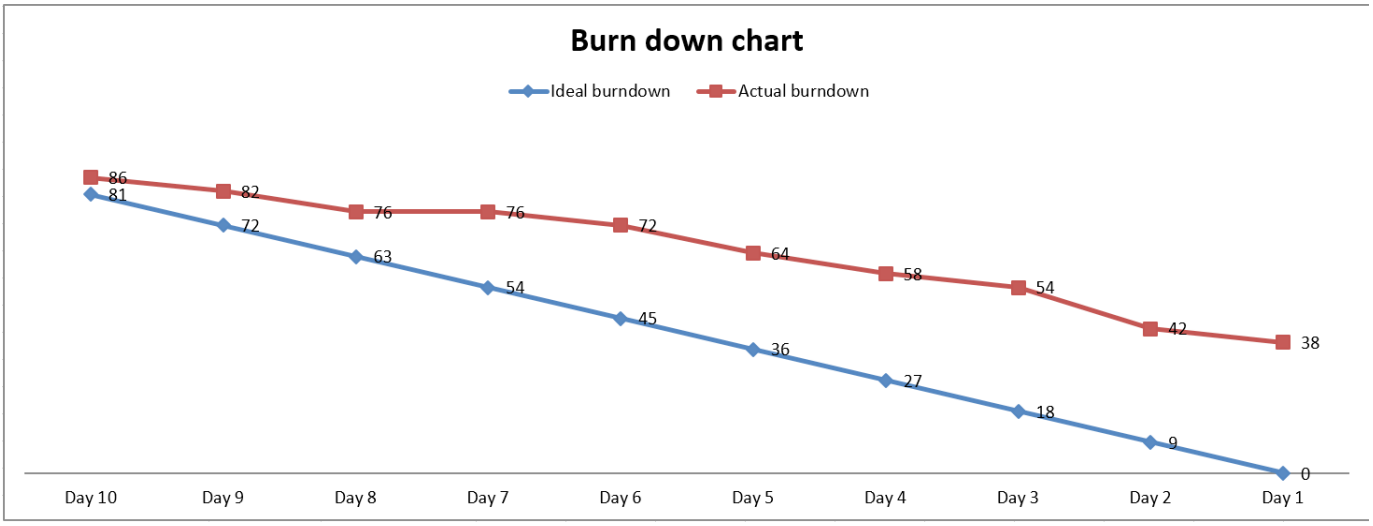

Figure 5.7: Burndown chart for Sprint 2

5.8 Sprint 3

The last sprint three was focused on the testing, debugging and design of the software. The group decided to spent the remaining time of the project implementing as many tests as they can before the deadline, without any estimated times. This decision was made due to uncertainty of how much the group can achieve before the end of the project. Meaning, planning poker was skipped and most of the classic sprint events were not conducted, leaving the sprint three with the minor goals of test and design improvements. The last sprint has focus laid on:

- Changing design of the system - placement, shape of buttons etc.;
- Implementing UNIT tests for the client;

Decision of skipping planning poker and leave the tasks without an estimates did not change the final version of the system. One of the minor problems that the group encountered was the determination of how many UNIT tests should be implemented, due to the big amount of the program, meaning not all UNIT tests were implemented, leaving the implementation of the rest to be done after the hand-in of the project.

Chapter 6

Conclusion

In conclusion, the project ended up as fully developed, there were, of course, difficulties on the way and many other futures could be added, but overall the team made great progress. Some of the difficulties we had are related with organizational matters and had an impact on the quality of the project, such as:

- The group did not have a specific Scrum master during every sprint;
  - Everybody contributed in every way possible, positions were switched and there were obvious moments when members acted as leaders, but no one had the responsibility of a Scrum master.

- As an after effect of the first matter came the unbalanced number of meetings the group had;
  - Sometimes there we too many meetings and not that much to work on and sometimes the number of meeting was not enough, leaving some of the tasks to be done at home and delayed from the schedule.

In opposition with these matters, some tools gave a tremendous help that kept the team on track and stay productive, for example:

- Planning Poker brought a great benefit by saving time in decision making moments;

- Burndown chart helped everybody know how the project is dealing with the planned deadlines;

- Also tools like Github and Visual Studio Team Service made it possible for everybody to see the status of the project at anytime along with good communication between group members.

Appendix A

1.1 Persona

Persona

John

Age: 20
Occupation: Student
Location: Aalborg, Denmark

Personal interests:
John is passionate about computers and technology in general. He aims to graduate a bachelor degree in Computer Science and become a professional software developer.

Lifestyle:
John is studying Computer Science and so he spends a lot of time at the office, raising his concern about how comfortable he is while using the computer. He likes to learn more and more stuff about computers whilst playing games, listening to music and being very active on social media. Rather than that, he is an outgoing person who enjoys to be around good company.

Area of concern:
John has a hard time studying at home, therefore he wants to move out in a new apartment and have a new home workspace where he can focus better on his studies. He plans to buy new furniture and get rid of the old one. He thought of selling the old furniture, but he doesn't know a simple and easy way to do that. Here is where our idea solves the problem and helps John sell his items through an app that is based on a fair auction system. The app can help him add as many items as he likes and specify a minimum or desired price at which every item should be sold, to fit his needs. If the final bid doesn't reach the price that John chose, he can decide whether or not he will the sell the item.

1.2 Group Contract

GROUP CONTRACT

Group Name	Group 3
Group Members	Alexander Zahariev Arnas Sokolovas Jindrich Jehlicka Marius Constantin Untaru
Project	Third Semester

As a group member, I will:

- Try to do my best working as a group member.
- Be understandable and helpful to fellow group member.
- Express my ideas and thoughts about the project freely without feeling that my thoughts can be silly or out of context.
- Listen and give honest opinion about other's assignments and ideas.
- Work on assignments cooperatively.
- Ask help from other group members if needed, while completing the tasks.
- Believe that I can and will learn.

As a group member, I will NOT:

- Object the decisions made in the group while I was inactive or absent.
- Leave/abandon my fellow group members who are struggling on assignments alone.
- Miss group meetings. In case of emergency I will give them a notice.
- Ignore the feedback and comments made by supervisor or fellow group members.

Signatures:

Alexander Zahariev _____

Arnas Sokolovas _____

Jindrich Jehlicka _____

Marius Constantin Untaru _____

1.3 User Stories

*As a *user type*, I want to *perform a certain action* in order to *achieve a certain goal*.'*

1. As a User, I want to bid on an item through an auction in order to buy it at a good price.
2. As a User, I want to have a single account in order to both sell and buy goods.
    a. As a user, I want to have a single account in order to sell goods.
    b. As a user, I want to have a single account in order to buy goods.
3. As a User, I want to have an easy way of selling my goods with a good price.
4. As a User, I want to sell the things that otherwise I would throw away in order to earn some extra money.
5. As a User, I want to have full control and a clear view of my account in order to not waste money and time.
6. As a User, I want to be able to see the goods in categories in order to easily browse them.
7. As a User, I want to be able to edit my account details if want to change or delete my information.
8. As a user, i want to be able to use a keyword, so that i can find the wanted item.
9. As a user, i want to see my current bids, so i do not forget what items i have bidded on.
10. As a user, i want
11. As a user, I want to find frequently asked questions to save time in case I have an issue.
12. As a user, I want to see the most popular auctions as I might be interest to bid too.
13. As a user, I want to be able to create a wishlist or save the auctions I am interest in in order to follow them.
14. As a user, I want to be notified by email whenever something happens to my auctions.
15. As a user, I want to be able to choose what personal information will be shown to the other users so that I can keep my personal information private.
16. As a user, I want to be able to get reviews whenever I sell something so that it will be easier for me to sell other goods in the future.

Bibliography

- Ambler, Scott; Constantine, Larry (2002). The Unified Process Transition and Production Phases. CMP Books. ISBN 1-57820-092-X.
- Ambler, Scott (2002). Agile Modeling: Effective Practices for EXtreme Programming and the Unified Process. J. Wiley. ISBN 0-471-20282-7.
- Bergstrom, Stefan; Raberg, Lotta (2004). Adopting the Rational Unified Process: Success with the RUP. ISBN 0-321-20294-5.
- Cohn, Mike. "Planning Poker Cards: Effective Agile Planning and Estimation". Mountain Goat Software. Mountain Goat Software. Retrieved 30 March 2016.
- "Planning Poker: An Agile Estimating and Planning Technique".Cohn, n.d. Mountain Goat Software. www.mountaingoatsoftware.com/agile/planning-poker [Accessed 18 Dec. 2016].
- "Unified process vs agile processes".Chris Collins, [online] Available at: https://ccollins.wordpress.com/2008/06/11/unified-process-vs-agile-processes/ [Accessed 18 Dec. 2016].
- Mike Cohn (November 2005). "Agile Estimating and Planning". Mountain Goat Software. Retrieved 2008-02-01.
- Larman, Craig (2004). Agile and Iterative Development: A Manager's Guide. ISBN 0-13-111155-8.
- Scott, Kendall (2002). The Unified Process Explained. ISBN 0-201-74204-7.
- Kruchten, Philippe (2004). The Rational Unified Process: An Introduction (3rd Ed.). ISBN 0-321-19770-4.
- Kroll, Per; Kruchten, Philippe (2003). The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP. ISBN 0-321-16609-4.
- K Moløkken-Østvold, NC Haugen (10–13 April 2007). "Combining Estimates with Planning Poker—An Empirical Study". 18th Australian Software Engineering Conference. IEEE: 349–58. doi:10.1109/ASWEC.2007.15. Retrieved 1 February 2008.
- Wenzel, Joel (December 2012). "Burn Down Chart Tutorial: Simple Agile Project Tracking".
- "A programmer's diary".Goodoldmanoj.com, n.d. Goodoldmanoj.com [Accessed 18 Dec. 2016].
- "Feel The Burn, Getting the Most out of Burn Charts - Better Software, Volume 11, Issue 5, pp. 26-31" (PDF). July–August 2009.

- “Planning poker - Trademark, Service Mark #3473287”. Trademark Status & Document Retrieval (TSDR). Jan 15, 2008. Retrieved 2014-05-26.

- “Planning Poker: The Best Agile Planning Tool”. PlanningPoker.com. Retrieved 30 March 2016.

- “The burn-down chart: an effective planning and tracking tool - scrum alliance”.Scrumalliance.org. (n.d.).www.scrumalliance.org/community/articles/2013/august/burn-down-chart-%E2%80%93-an-effective-planning-and-tracki [Accessed 18 Dec. 2016].

- “Waterfall vs. Agile: which methodology is right for your project?”Segue Technologies.www.seguetech.com/waterfall-vs-agile-methodology/ [Accessed 18 Dec. 2016].

- “What is unified process (up)? - definition from techopedia”.Techopedia.com. www.techopedia.com/definition/3885/unified-process-up [Accessed 18 Dec. 2016].

- “What is agile? (10 key principles of agile) | all about agile”.Allaboutagile.com. www.allaboutagile.com/what-is-agile-10-key-principles [Accessed 18 Dec. 2016].

- “What is quality assurance? Definition and principles for projects”Project-management-skills.com. (n.d.). www.project-management-skills.com/what-is-quality-assurance.html [Accessed 18 Dec. 2016]. (PDF). July–August 2009.