Navneet Chamala

nach14@student.bth.se

Team Number-6

PA-2515 Applied Software Project Management

Individual Report

Answer 1- Life Cycle Model

1.1 Life Cycle Model- We have chosen Scrum software development with pair program technique as our software development lifecycle model. In the first kick-off meeting, we discussed our technical and managerial skill sets. Among the 8 group members, 7 of them did not have any prior real time experience and one of the team members (Frédéric Moreira) had 6 months of work experience. However, Frederic had no experience with software project management. The primary reason for choosing scrum is that we can review the product development at the end of each sprint using a project deliverable which gives us a clear understanding on the status of the project and how to plan the tasks in the further sprints[1]. Also, the pair programming technique will help the team members to share knowledge and learn from each other[2]. Scrum also facilitates changes in requirements and is more flexible when compared to the other traditional methods like waterfall model. In the present context, where the requirements aren't fixed scrum was the best choice we could make. Since, all the team members had different levels of skill sets, using scrum made it convenient to distribute tasks as the members of the team volunteered in selecting the functionality based on their own capability. By combining these two techniques we intended to develop the website. Some of the team members have already informed the team that they won't be available for few days during the course of the project. Helen, one of our team members, though not experienced in implementing the scrum model was well aware of how the model functions and explained the team members the activities carried out in the development life cycle.

1.2 Customer's role: The customer plays a crucial role in the scrum development lifecycle. At the end of each sprint the customer reviews the progress of the project and suggests any changes or might even give new requirements. In agile methodologies like scrum, the user and analysts exhibit equal responsibility in deciding the requirements [3]. Like mentioned above, we have taken up the pair programming technique where one of them develops and the other programmer reviews. During reviewing the code the programmer also checks it from the customer perspective. This was not done explicitly but was reviewed implicitly among the pairs. Thus, the whole team plays the role of a customer, though we haven't defined a specific role called customer it was implemented implicitly.

1.3 Strengths and weaknesses: The strengths and weaknesses of the chosen model from my perspective are, there were changes we made to the initial project during the course of the project. We anticipated this and we have designed our plan of action and this is one of the areas where scrum model was really helpful. With the help of weekly sprint meetings the whole team was aware of the status of the project. Like mentioned above there was a case where the team members were not available but still we could manage and continue the development process. Re-incorporating a team member when he re-joined without much was another advantage where pair programming was helpful.

Some of the disadvantages were, since none of us had prior experience, it was difficult to execute and implement all the activities of scrum. Daily scrum calls were not possible because the team members also had other courses and priorities and it was not possible to schedule a meeting every day. Task allocation among the team was another drawback as the tasks are distributed based on the willingness of the developer. We instead planned to have daily scrum calls among the pair programmers alone which did not

turn out to be enough productive. I did not find pair programming technique as effective as I thought it would be, since we decided that the pairs are grouped based on their home country so that there would be no communication gap among the team members. But grouping teams based on that criteria did not help in sharing experiences.

1.4 Re-use Lifecycle model: I would definitely recommend the same life cycle model for a similar project in the future. Scrum, like already mentioned accommodates change in requirements and is more flexible when compared to the other traditional models. Regular scrum meetings involving the whole scrum team helps the team to have a complete overview of the development process and the status of the project. The team members can also select the tasks based on their own skillsets and interest. However, there were certain activities where we couldn't execute as planned and I would do it differently the next time. We couldn't implement daily scrum calls properly and I will want to execute those calls more effectively. The pairing of programmers could have been done in a better way rather than grouping them based on nationality. Scrum master was selected based on programming experience which we realized was not ideal during a later stage of the project. Hence, if I were to implement the same developmental model I will select the scrum master based on not just their programming experience but also their knowledge about software project management using scrum, their experience in working in a scrum environment. These are the changes that I would implement the next time I use scrum for a similar project.

2. Schedule and Effort Management

2.1 Process, Methods and tools used to measure and track progress: Based on the scrum features, we have initially developed a product backlog which charted the details of all the functionality that we have intended to develop. We maintained the project backlog in a spreadsheet where every function had a unique reference id, clear description of what it does and how it can be implemented. From the product backlog we took few functions based on the order of priority and formulated the sprint backlog. The sprint backlog listed the functions and the person responsible to develop the function. The sprint backlog was colour coded with three colours defined i.e. red, green and yellow. If the function is denoted by red it means it hasn't been taken up yet. Green denotes completed and yellow denotes currently developing. This made it easy for us to know which functions are done and which of them are pending. This gave us a clear picture of the status of the project. We had a time log wherein we used to enter the total number of hours worked on a particular day and what we have done on that particular day. This gave us an idea to who has done what during a day. With the help of the time log we developed project burn down charts and graphs to keep a check on how we are functioning with respect to time. Graphs were plotted sprint wise and also with respect to the time available and time spent (see Appendix). This helped us in tracking our progress in terms of time. We also tracked the progress based on functions. With the help of graphs we tried to illustrate the number of functions and how many have been started, finished or cancelled. (see Appendix).

2.2 Deviation from team's project plan: In the initial stages we planned and estimated the time and effort for 8 people. One of the team members has left without any notice and so we re-estimated everything for 7 people. In sprint 2, unfortunately another team member had to leave because of a family emergency and we were not sure when or whether or not he is going to be back. Hence, we re-estimated time and effort for 6 people. The first two sprints we were on par with the planned estimation. In sprints 3 and sprints 4 the team couldn't invest the required number of hours that we were supposed to. The original plan was to finish the development within four sprints and spend the vacation for writing the individual reports. But due to the delay that occurred during sprint 3 and 4 it wasn't possible. In sprint 4, one of the team members re-joined the team and then we had to again re-estimate time and effort for the remaining sprints. As we couldn't finish the project by December 19, we included three more sprints to our existing project plan. In sprint 5, we decided that we weren't able to develop some of the functions. So, in order to keep up with the scope we introduced new functionality and developed those. We also planned usability and automated testing which did not happen. However, as planned we conducted acceptance testing for all the functions. The roles and responsibilities also changed along the course of the project. Initially Frederic was named scrum master but after he left, the team decided Hamid would be the scrum master from sprint 2. In sprint 6, Hamid was not comfortable with the position and Helen had taken up the position. The pair programmers have also changed along the course of project development.

2.3 Reasons for deviation: One of the main reasons for deviation is that the lack of experience in developing made it difficult for the team members to learn and implement simultaneously. It took us time to learn and this delayed the delayed the implementation process. However, the fact that the team members were less motivated during some stages of the project which had an impact on our work to the

fullest potential cannot be denied. The reason for low motivation according to me was less team cohesion. The team did not really know much about each other. Trying to know each other more would have created a friendlier working environment. Scrum masters have been changed twice during the project. The first time it was because the scrum master left the team without proper notice. Second time i.e. in sprint 6, Hamid was not comfortable with the position as he was also handling most of the development part and Helen was made the scrum master.

2.4 Compensating deviation: After sprint 1, we re-estimated time and effort for a total of 7 members and after sprint 3 when another team-mate left we re-estimated the effort to 6 people. When one of the team members returned we again estimated the time and effort for the remaining for 7 people. Whenever there was a need to change the scrum master it was done both voluntarily and on basis of team's decision. Nobody had any issues with the responsibilities assigned to them. Attempts were made by team members to mingle with the team for building cohesion among the team but went in vain as it wasn't that helpful. We have depicted the deviation by plotting a graph showing the deviation. (see Appendix)

2.5 Scope for Improvement: Based on the experience I've had from this project, in future I will take assurance from the team members and clear information regarding their availability. However, there is nothing we could do in case of an emergency. However, I would make sure that the planning is flexible enough and the chosen process model can accommodate the changes. In order to improve team cohesion I would have more interaction sessions at the beginning of the project and in between the project as well which would motivate the members of the team to deliver better.

3. Risk Management and Quality Management

3.1 Risk Identification: Initially after the kick-off meeting and deciding on scrum as our process model, in the first sprint we were to have a plan of action. There were a set of responsibilities that had to be assigned and one of them being risk management. I volunteered to take up the task of risk management. In the first sprint meeting, all the team members have shared their previous experiences from their respective student projects. After the first sprint meeting, based on the team members and my previous experiences and their suggestions I have identified a set of possible risks that might occur during the course of the project. All the risks identified were registered in a risk registry. Risk registry holds details of the risks identified the impact, effect, avoidance and the mitigation strategy of the risk. Identifying and managing risks was a continual process throughout the project. During the course of project development, the risk register was updated at the end of each sprint. Risks are added to the existing risk registry if there were any new risks reported by the team members during that sprint. Along with the new risks, the old ones are also tracked and updated every sprint.

3.2 Risks materialized: From the initial risks we have identified there were several risks which happened to be true. From the initial 8 risks that we have identified, 6 risks were materialized. The risk and the mitigation strategy we have implemented are tabulated below.

Risk ID	**Risk**	**Mitigation Strategy**
R02	Lack of programming experience.	The team members who are experienced and efficient with the programming language helped in solving issues related to programming.
R03	Unavailability of team members	Some of the team members were unavailable and their work was taken up by another team member or mostly by his partner in pair program.
R04	Issues in team dynamics	There were issues now and then among the team members which arose due to difference of opinion. However, none of them were that serious and a simple discussion used to suffice.
R05	Missing out on regular scrum calls and meetings.	Regular scrum calls were not possible among the whole team, so daily scrum calls were organized among the pair programmers.
R07	Delay in finishing a task	We have allocated some spare or extra time during our initial project plan and this time was utilized here.
R08	Change in requirements	We made sure that the changes in requirements are within the scope and achievable.

Table 1: Expected materialized risks

3.3 Unexpected risks: Apart from the above stated risks there were few other risks which have taken light during the course of the project. Some of these risks were of little severity while some of them were severe. In table-2, I give an account of all these risks with a reason why we couldn't anticipate them before.

Risk ID	Risk	Reason
R09	Loss of scrum master	The initial scrum master informed at the end of sprint 1 that he wouldn't be available for few days during the project and after sprint 1 we did not hear from him so we did not wait for him and appointed a new scrum master.
R10, 14	Issues in dealing with webmatrix IDE.	Some members had difficulties in using the IDE which was unexpected as we initially thought its functioning seemed easy.
R13	Lagging in final product development	There were few functions which took more time than expected and also all the users couldn't invest the required amount of time. We were a bit over ambitious not to consider this risk initially.
R15	Difficulty for the team member who re-joined to understand the status of the project.	We expected the unavailability of team members as a risk but failed predict that it would be difficult to re-incorporate the team member back into the team due to lack of experience.
R16	Changing the scrum master.	After changing the scrum master in sprint 1 we again took into consideration the programming experience of the team member for deciding the scrum master and this is where we went wrong, so we had to change the scrum master again.

Table 2: Risks that materialized during project development

We failed to predict the above set of risks during the initial plan due to lack of experience among the team members.

3.4 Quality criteria: Due to lack of experience, we were incapable of properly defining a quality criterion. However, initially the plan was to consider lines of code and comment percentage of source code as our quality criteria. This criterion was later changed and test plans and test results have become the basis for deciding the quality of the product.

3.5 Quality control and Quality Assurance: Quality control was done by using three processes; they are (i) Meetings, (ii) Reviews, (iii) Defect/Issue tracking and changes. We had daily meetings occasionally among the pair programmers and sprint meetings with the whole team at the end of each sprint. In these meeting the team members have a look at the progress of the project and ideas are shared among the team to improve quality. Reviews are other methods we followed within the pair programming teams as it is not possible for all the team members to review the work to improve the quality of product. By using defect tracking we check how the website is functioning. We check this by giving a few inputs to the system and see how it responds and if there is any unexpected behavior. However, by the end of the project we couldn't efficiently check the quality of the product.

4. Project Plan

4.1 Positives from team performance: This was an intriguingly new experience for all the team members. The team initially had 8 members with team members coming from different countries and different cultural backgrounds. It was a great learning experience to all of us. Here I list those aspects of the project plan where we as a team have performed well.
One of the important things was to re-incorporate the team member who was back in the team after few days. It was difficult for him to initially track the status of the project. His pair programming partner and the rest of the team members helped him out in understanding the status of the project. This is something that we have done as a team and did it efficiently.
Though, we couldn't really organize daily scrum calls we have collectively organized scrum meetings every week to discuss the status and progress of the project. There were times when some of the team members were not able to attend the meeting in person because of health issues but they made themselves available over skype and participated in the discussion. This shows that the team members were really very committed towards the project.
The team never really had strong issues or conflicts. There were difference of opinions sometimes but we were able to manage and act accordingly.

4.2 Things to improve from team performance: There were also hindrances that we have faced during the course of the project. In the end, after retrospection of the team's performance there are certain things that I personally feel we could have done better. In this part of the report, I provide a list of those activities.
We could have organized the daily scrum calls more effectively. If not in person, we could have used online media like google hangouts, skype etc. to organize short meetings daily.
In the initial kick-off meeting, we have only discussed about the programming and technical skillsets of the team members but it would have been better had we discussed about each other's interests and get to know more about each other. That would have helped in better working conditions than what it was.
Effort and time could have been better estimated than how we executed it.
We couldn't explicitly test the functionalities at the end of each sprint. However, we tested the functions in the last sprint where we called it the testing week. This is not how it is done in scrum and this is something where we could have performed better. Explicitly testing the system at the end of every sprint would also improve the quality.
We haven't really documented to a great extent. It would have been more efficient had we documented the development process. By documenting the information about key functionality and other activities of the project it will be easy to re-use this process model in the future.

Thus, these are the positives and drawbacks that I gained from this project.

5. My Role

5.1 Role: Almost every member had to manage multiple roles and so did I. Initially, I was responsible for designing and implementing the main layout of the website. I was also responsible for risk management. I volunteered to take up the above two roles. Once I was done with developing the main layout of the website, I have also taken up the role of developing the addFood, addMenu, viewFood and viewMenu pages. I also played a key role during the initial meetings when the functionality of the project was being discussed. I have provided all my ideas to the team. Throughout the project, I was pair programming with M.V.N.A Uday.

5.1 5.2 Knowledge practiced and gained during the project: Yes, the above mentioned roles were the most effective use of my skills and knowledge. I learnt HTML, CSS and jquery programming languages and I'm good at it. This motivated me in volunteering to design the main layout of the website. By doing so, I gained experience in using those languages and also I could revise the concepts of the aforementioned programming languages.
Also, we developed the code on an IDE called webmatrix. It was a learning experience for me as this was the first time I used this IDE. I have also taken up the responsibility to manage the risks. I volunteered to take up this role because I personally fancy analyzing challenges and providing a solution to it and it was a great opportunity to use that skill in this project.
I have also learnt ASP .Net before but had little experience in implementing it. I have gained experience in accessing the database using ASP .net webpages. I have also gained experience in using the SQL server database. Thus, from this project I could present my technical skills and learnt the above stated few technical concepts as well. Along with the technical skills, I have developed and learnt a lot about managerial activities. I realized from this project that, coding is not the only important aspect of developing a software project management. There are a lot of activities other than coding like scrum meetings, risk management, requirement analysis and implementing that form into developing a software project.
This is the first time I was part of a Scrum team and the project helped me in gaining knowledge related to the various activities in scrum and how they are implemented. Thus, the project was useful in presenting my technical skills and also helped me to a great extent in improving my managerial and technical skillsets.

5.3 Contribution: Like mentioned above, I am good at developing the front end or the client side programming using HTML, CSS, Jquery and Javascript. Thus, I made sure that I develop the main layout on time. We all know that, the main layout is like the foundation of building a website. Hence, it was important for us to develop it on time so that the remaining functionalities can be implemented. I developed the main layout on time in sprint 1.
I was also managing the risk registry and updating the risk factor of the development process and also letting the rest of the team members know what could be done in order to avoid unexpected situations. I also played a key role in voicing my ideas and opinions during the sprint meetings. This was very useful in deciding the functionality and the process. I tried to rectify team member's problems whenever I was aware of the solution. Thus, I made sure that I contributed to the fullest extent possible.

5.4 Scope for improvement: Even though I tried to give my best in the project there were certain things I couldn't execute properly and if I were to perform the same roles again I would want to work on the following things.

Firstly, I did not spend all my hours allocated for this project. By the end of the project there were around 15 hours remaining. I couldn't work to my full quota of time as I was procrastinating work because I had other pending assignments and I couldn't manage both the tasks simultaneously. I would want to change this in my future endeavors by concentrating on a single task at a time without multitasking.

I was over ambitious while taking up the functionality I would want to develop and ended up not being able to do it. This resulted in delay of developing a function. In the future projects I would make a SWOT analysis to identify my strengths, weaknesses, opportunities and threats before choosing a task.

There were a couple of occasions where I was late to the scrum meetings and I would want to avoid that in the next future projects.

References

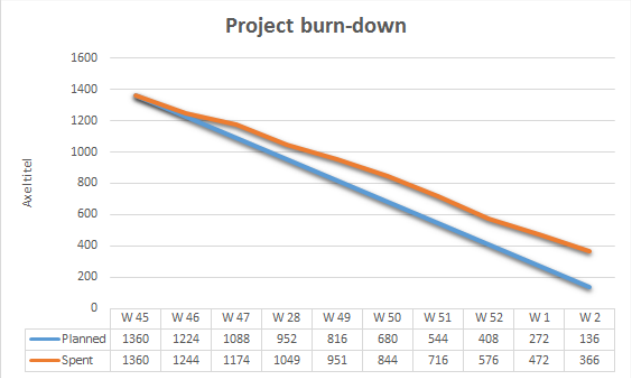
[1] G. Rodriguez, A. Soria, and M. Campo, "Teaching Scrum to Software Engineering Students with Virtual Reality Support," in *Advances in New Technologies, Interactive Interfaces and Communicability. Second International Conference, ADNTIIC 2011, 5-7 Dec. 2011*, 2012, pp. 140–50.

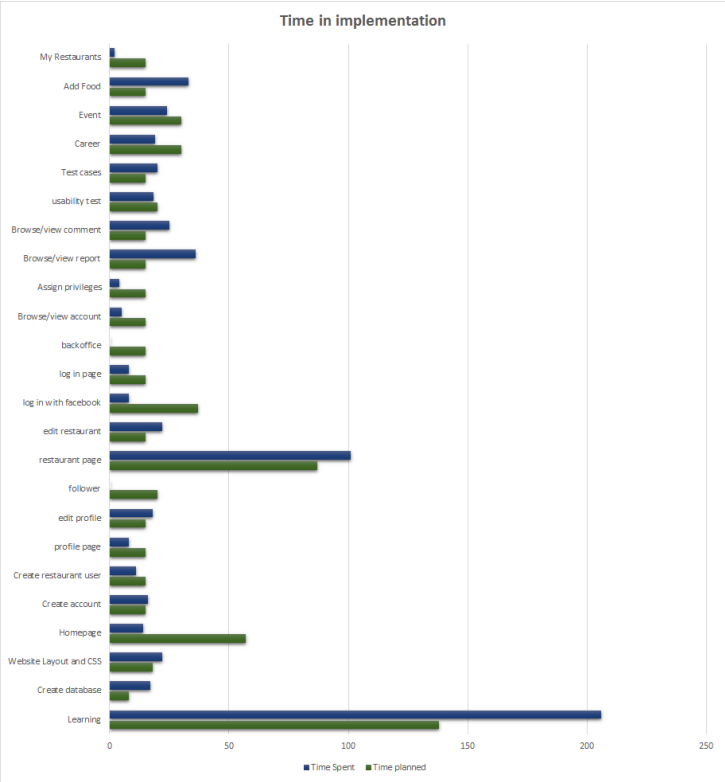
[2] G. Braught, L. M. Eby, and T. Wahls, "The effects of pair-programming on individual programming skill," *SIGCSE Bull.*, vol. 40, no. 1, pp. 200–4, Mar. 2008.

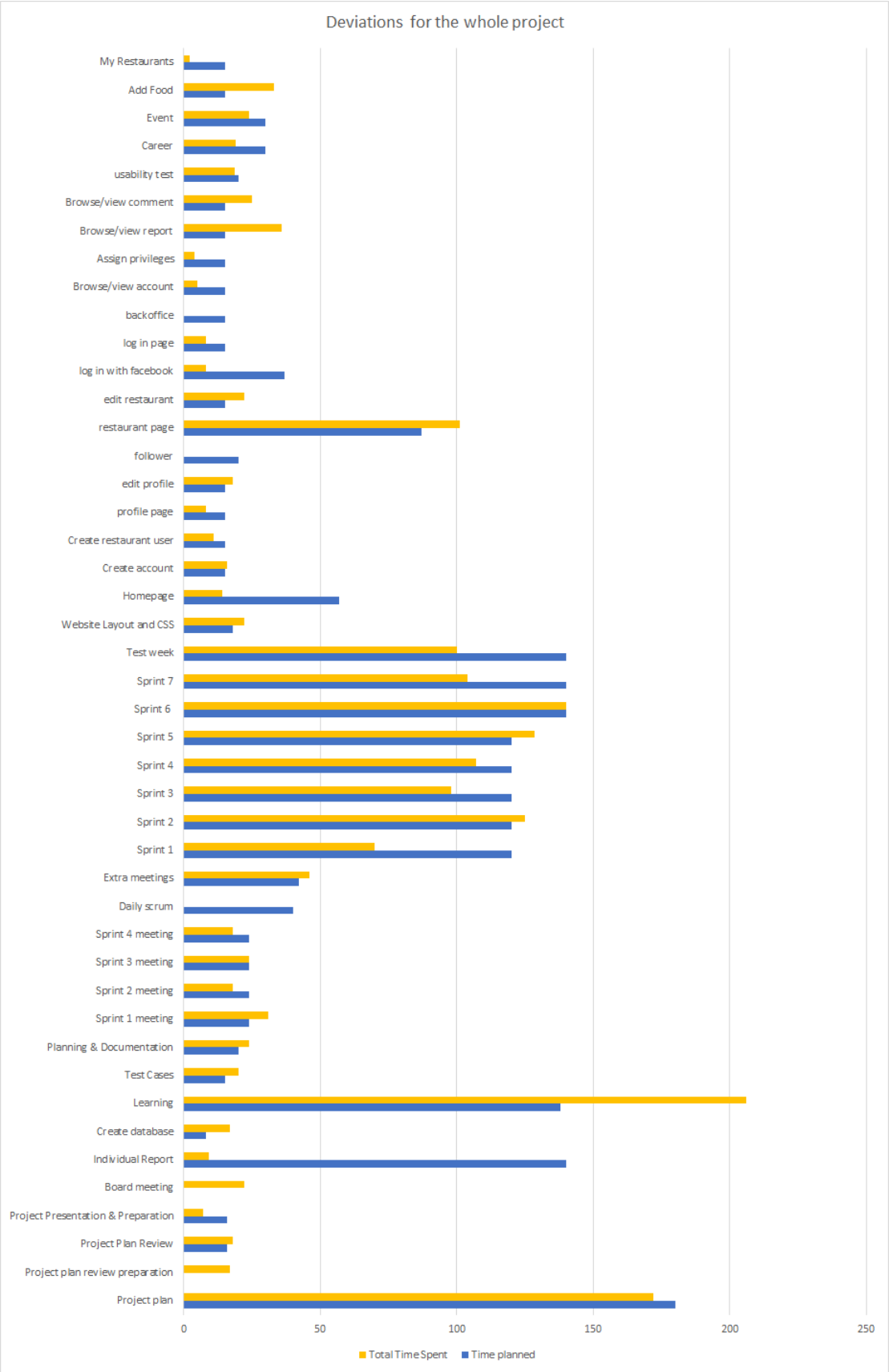
[3] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Inf. Softw. Technol.*, vol. 50, no. 9–10, pp. 833–859, Aug. 2008.

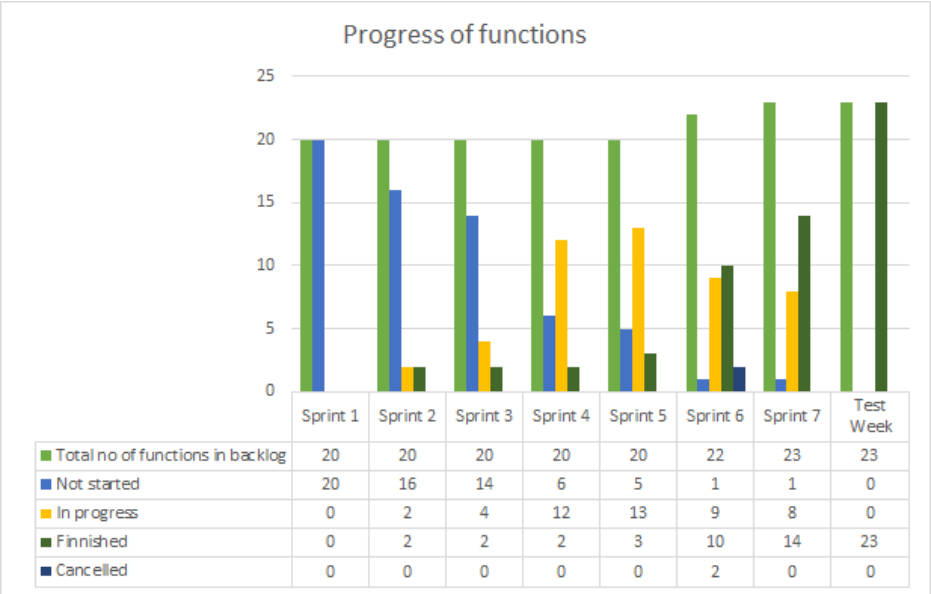
Appendix


Time available vs time spent
300
250
200
150
100
50
0
Axeltitel
Pre-phase
Sprint 1
Sprint 2
Sprint 3
Sprint 4
Sprint 5
Sprint 6
Sprint 7
Test week
Presentation week
Planned 240 120 120 120 120 140 140 140 140 140
Spent 167 70 125 98 107 128,5 140 104 106

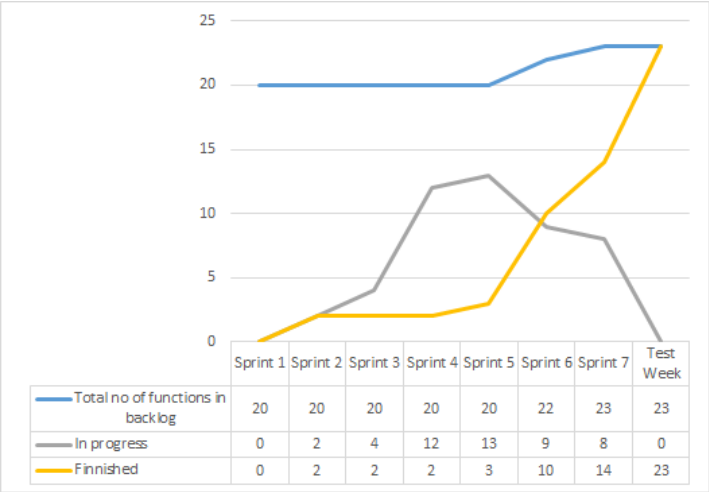

Project burn-down
1600
1400
1200
1000
800
600
400
200
0
Axeltitel
W 45 W 46 W 47 W 28 W 49 W 50 W 51 W 52 W 1 W 2
Planned 1360 1224 1088 952 816 680 544 408 272 136
Spent 1360 1244 1174 1049 951 844 716 576 472 366


Time in implementation
My Restaurants
Add Food
Event
Career
Test cases
usability test
Browse/view comment
Browse/view report
Assign privileges
Browse/view account
backoffice
log in page
log in with facebook
edit restaurant
restaurant page
follower
edit profile
profile page
Create restaurant user
Create account
Homepage
Website Layout and CSS
Create database
Learning
0
50
100
150
200
250
Time Spent
Time planned


Deviations for the whole project
My Restaurants
Add Food
Event
Career
usability test
Browse/view comment
Browse/view report
Assign privileges
Browse/view account
backoffice
log in page
log in with facebook
edit restaurant
restaurant page
follower
edit profile
profile page
Create restaurant user
Create account
Homepage
Website Layout and CSS
Test week
Sprint 7
Sprint 6
Sprint 5
Sprint 4
Sprint 3
Sprint 2
Sprint 1
Extra meetings
Daily scrum
Sprint 4 meeting
Sprint 3 meeting
Sprint 2 meeting
Sprint 1 meeting
Planning & Documentation
Test Cases
Learning
Create database
Individual Report
Board meeting
Project Presentation & Preparation
Project Plan Review
Project plan review preparation
Project plan
0
50
100
150
200
250
Total Time Spent
Time planned



	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Sprint 7	Test Week
Total no of functions in backlog	20	20	20	20	20	22	23	23
Not started	20	16	14	6	5	1	1	0
In progress	0	2	4	12	13	9	8	0
Finnished	0	2	2	2	3	10	14	23
Cancelled	0	0	0	0	0	2	0	0



	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Sprint 7	Test Week
Total no of functions in backlog	20	20	20	20	20	22	23	23
In progress	0	2	4	12	13	9	8	0
Finnished	0	2	2	2	3	10	14	23