INDIVIDUAL REPORT: APPLIED SOFTWARE PROJECT MANAGEMENT

Sai Priyatham Dongoor

Sado16@student.bth.se

Team: 2

QUESTION 1: PROJECT DEVELOPMENT METHODOLOGY

1.1 Which project development methodology is used by your team?

- The software development methodology used in our project is SCRUM methodology.
- The main motive for choosing this methodology is, it delivers a quality working product in iterations. Risks in the project can be detected and mitigated easily in this methodology as each working feature of the product is developed as a task set in sprints, which are tested immediately after finishing of each sprint.
- The features of the software to be produced is divided in to small tasks and set of tasks are executed in each sprint.
- Sprint retrospective meeting's as a part of scrum methodology is helpful for discussing any faults or issues among team members and can be solved accordingly.
- Progress of the software development can be measured by using burndown charts in this methodology.

- Moreover, in survey we have conducted for an apt software development methodology for our project, we found that SCRUM methodology is very apt for our project (wow-weather: small scale web application).

1.2 Adaptations on SDLC model made by your team?

- In SCRUM methodology, a scrum master is responsible for facilitating meetings, assessing product backlog, making sure that things are working as per SCRUM methodology. He does not participate in development or testing of the product. **Where as in our project**, due to availability of few team mates (7 members) we felt that it was unwise to assign a team member, only the role of scrum master, instead in our project scrum master also participated in development process also. **Yes,** It worked pretty well for our project as we had only 7 members to complete the project.
- In scrum methodology, scrum meetings are held daily without fail, where issues, innovative ideas regarding development process is discussed among team members, **in our project** as it is academic project, due to other subjects, we conducted 2 to 3 sprint meetings per week, by considering team member's schedules. This adaptation did not affect us negatively, instead this helped us in completing this project along without effecting other subject assignments.
- In regular scrum methodology, technicians who lack the skill set are trained before starting of the project, but **in our project** as it is academic project and time for developing the product was limited, team members who lack the coding skills trained themselves (with team members help) before developing each sprint or improving themselves simultaneously while developing. **Yes**, this adaptation helped us a lot in avoiding waste of time as other teammates who has good skill set, started development process while guiding others with low skill set in parallel.
- Due to time constraint all the features stated in project plan could not be completed on time, so few features were omitted from the product backlog. Apart from features, design of the produced product is also different from the design which was made as a blue print and before starting of the project. This is because of considering ease of developing a feature. This adaptation was important to us as due to time constraint we have developed few features as quickly as possible, and developing those features strictly as per our design would result in a lot of effort.

1.3 Strengths of the software development methodology used by our team

- As SCRUM methodology which was used by our team is an iterative process where features or requirements of the software to be produced is divided into a set of tasks and are assigned to team members, adding or deleting any feature is very easy and does not require a lot of effort, where as it takes lot of effort in other traditional models. For example, in our project due to time constraint and difficulty in developing we have deleted feature like showing searched city or place location on google maps.
- As SCRUM model facilitates project monitoring through sprint burndown charts, it helped us track our progress and improve our selves if we are going out of track.
- Features as a set of tasks which were assigned to team members were listed in a doc file in google drive along with number of hours spent and percentage of task

completed, which helped us in tracking the progress of a feature by the end of each sprint (as few team members have low technical skills they couldn't complete few tasks by the end of sprint, but completed in few days after the sprint). More over this also helped us in tracking which task is assigned to whom, so that no task is developed by 2 team members separately.

- Sprint meetings as a part of SCRUM methodology helped us in discussing issues occurred in developing process among team members and depending on the suggestions given those issues were resolved and also assigning tasks to team members before starting of the sprints, which is done by scrum master.

1.4weakness of the software development methodology used:

- Usually scrum project is driven to success by an experienced scrum master, but as our project is academic project, we didn't have experienced scrum master, which is the reason why the task distributions and completion of tasks by team members were not done precisely in our project. Especially in our last sprint where time constraint made only few team members work on most of the development process (as others who have less skill set were working slowly on the tasks assigned to them). Approaching deadline for each sprint has developed pressure on team members which further worsened the development process. This resulted in failure of team members in the course. I guess this would have not occurred if we used some other traditional model such as *waterfall or incremental model*, as development process is done at once in those models which might result in low pressure on team members in completing tasks.
- In case of depending tasks such that completing one task which is dependent on completion of other task might cause dead locks, if those 2 tasks are assigned to 2 different team mates (one task (dependent task) to team mate with good skills and task to be completed first to the team mate with low skills). Team mate with good skills can work on other task until then, but if the task done by teammate with low skill tends to fail, then it causes a lot of problems. In our project this problem has occurred and the one with good skills took over both the tasks and developed them. This problem would not have occurred in traditional models as tasks are not developed in parallel, instead they are developed in a sequential manner.

QUESTION 2: MONITORING AND CONTROL

2.1 What processes, methods, and tools did your team use to measure and track progress during your project?

- **Burn down charts** as a part of scrum methodology, is used to track our progress, deviation between ideal line and actual line show us if the project is going in correct direction or not. This helped us in tracking the process and correcting ourselves if project is not going as expected. These are drawn in excel. Brun down chart is the work left to do versus time graph, where backlog is on vertical axis and time is on horizontal axis.
- **Regular scrum meetings** helped us in monitoring and helping team mates with issues by suggesting them some improvements by discussing among ourselves. Scrum

meetings were held twice or thrice a week, by considering the availability of all the team members. Before beginning of each sprint, tasks were assigned to team members in these meetings by scrum master.

- A **task board** was uploaded on google drive where tasks assigned to each team member along with percentage of completion of task, number of hours spent was updated by each team member after completion of each task. Completion of task means the task is developed and tested. This helped in knowing who is responsible for the particular task and how much work is completed.
- Code written by each team member is uploaded in **GitHub**, and documents written are uploaded in **google drive** where work done by each team member can be accessed by every team member and changes can be made depending on the suggestions of other team members.

2.2 To what extent did your team's actual progress on the project deviate from your plans?

- Some features couldn't be finished by the end of sprint, which were finished few days after the end of sprints. Team mates who were slow in development were assigned less number of tasks to complete, which in turn affected negatively to those members. Pair programming helped teammates with low skills, but even pair programming was not done in correct way. This happened to us in few features like: in about screen which should be developed in sprint 1 but was completed in/during sprint 2; read latest location from search history from sprint 2 which was continued in sprint 3; we paused our second sprint for 5 days as we had to improve our project plan for this our sprint was rescheduled.
- Few features were developed differently from planned design such as: list of recent searches should be shown when we click the button for recent searches from sprint 2, but we have developed it a bit differently, they were shown on left side of the web page. Same with list of favourite searches from sprint 3.

2.3 Reasons for these deviations?

- First deviation was due to lack of coding skills among team members, only 3 team members from our team were familiar with developing web pages, or developing software products, others were new to these type of projects, all they have is conceptual knowledge but no practical knowledge. They had to learn and work on development of the product in parallel. Because of this few tasks which were dealt by team mates with low coding skills couldn't finish on time. Such tasks were completed during next sprint, and person responsible for that task should complete it and start working on his task from sprint 3
- Due to cultural differences, we hesitated to form pairs such that a team mate with good coding skills team up with the one with low coding skills and work together by helping each other.
- Due to lack of time, i.e. approaching date for final presentation. Our team members had to fasten development process, where actual design was neglected a bit and more effort is applied to include all or most of the features stated in the project plan. This happened as the sprint 2 was paused to work on project plan.

2.4 What actions did your team take as a result of these deviations? Explain why these particular actions were taken.

- The team mates who were slow in developing the product were given less tasks for developing, where others were completing tasks quickly, but this affected us negatively as we couldn't prove that we had worked 150 hours per person. We felt that completing the product was the main criteria to pass the course, this is what made us to assign fewer tasks to teammates with less development skills. We helped ourselves in development process by suggesting changes and helping with coding.
- This second deviation did not cause us any potential problem, as at the beginning of the project itself we discussed that the design can be altered depending on the execution of those features. So, it was fine with us, more over our project was successful as we succeeded in adding all the discussed features for our application.

2.5 As a result of your experience on this project, what would *you* do in future projects to measure, control, and react to deviations between actual and risksned progress?

- From the experience of this project it became clear that for pair programming we should plan and pair a team member with good coding skills with the one who has poor coding skills. Due to cultural differences, we hesitated and did not follow this correct procedure of pair programming. Which will not be repeated in my upcoming projects.
- As we have worked on this project, we gained some experience in developing a product. Altering initial design of any feature for better performance can be done, but we must be sure that the developed product is better (having all the important features which might lead the product to success) than the proposed design. This can be assured by discussing with team members and team leader, considering their suggestions necessary actions can be taken.
- Team member's true potential should be known and necessary training should be given before the start of project do avoid delays in development process. Cultural differences should not be allowed to interfere with the project as it ruins the working criteria between team members.

Question 3 - Risk Management and Quality Management

3.1 What processes, methods, and tools did your team use to identify the risks to the project and monitor risks during your project?

- Risks which might occur in our project were anticipated by our team mates in a scrum meeting and stated them in our project plan, fortunately risks which we have faced during our project were all among predicted and were documented during development of the project. As we have analysed their mitigation strategies also, we made sure that all the risks occurred were mitigated. Few examples of risks we have encountered during our project were stated in answer for the next question.
- Burn down charts which were drawn in excel sheets showed us our progress comparing with the ideal line, if there were any deviations, risks which lead those deviations were discussed and made sure that they all were mitigated.
- Status meetings were held if any risks occurred and what is to be done is discussed among our teammates and is done to ensure the success of our project. An example is stated in the next answer.

3.2 Did any of the risks anticipated in your project plan occur during the project?

Yes, we have faced few risks which were stated in the project plan, more over we haven't faced any risk apart from the ones we have documented in project plan. As we had predicted majority of the risks, we could avoid impacts of those risks. Few examples of risks which we have faced during our project and how did we mitigate or avoid those risks are stated below:

- **Risk 1:** team member may fall ill during development process and cannot work. In such cases as we discussed, the one who fell ill will mention his or her position to other team mates as soon as possible via WhatsApp, if that person cannot work then the task he is working on will be assigned to others or he will work on it after he gets better. **Yes,** this risk had occurred to us, one of our team mate was ill and couldn't work for 2 days, but since the task he was dealing with does not have any dependencies he finished that task by himself after he got better, because of this delay, one of the task assigned to him in that sprint was delayed and was finished after the end of that sprint.
- **Risk 2:** team member for some reason may not be in town and couldn't work, in such cases, the one who is away informs other team members, if he can finish the task in time or not. If he can, then his task can be assessed by us through GitHub or google drive, or if he can't work, then his tasks will be completed by others. **Yes,** this risk was encountered to us a few times, while Christmas 3 of our team mates were out of town for a week, due to personal problems, 2 of our team mates went to their home town for almost 10 days, but as they all accepted to still work remotely, their part of work was uploaded to GitHub and was informed to others via WhatsApp, if any suggestions were given to them regarding their tasks, they improvise their code and uploaded it. Thus this risk was handled.
- **Risk 3:** where team member cannot attend the scrum meeting, **Yes,** this risk was encountered few times, if a team member was out of town, or was busy with something else he couldn't attend the meeting, where he would join the meeting through call or he would collect information of what was discussed in the meeting from other team mates. Mostly we tried to organise meeting when everyone is available but sometimes we couldn't avoid this risk.
- **Risk 4:** this risk is the most crucial risk which we had, Team member lacks the development skills, to avoid this risk, we performed pair programming, but due to cultural differences, we didn't perform pair programming well, this affected our course grade. Our mitigation plan for this risk was good, but we didn't implement it in correct procedure. It was discussed in **2.5** due to size constraint I cannot explain it again here.

3.3 Did any risks or challenges occur that were *not* anticipated in the project plan?

- Fortunately, **No,** all the risks which we had encountered in the project were all anticipated by us, before the start of the project itself. **The risk which was not anticipated was**, not performing pair programming correctly. We assigned a team member with low skill with the team member with average skill, but not with the one with very good skill. None of our team mates were very good at coding, we all were new in developing web pages, few of them worked on android applications, but had very less idea on developing web pages. There were also team mates who had only conceptual knowledge, but no real time experience, which made our pair

programming weak. This affected us very badly in project grade as this lead to burden on team mates to complete tasks.

3.4 What processes, methods, and tools did your team use *both* to ensure *and* to assess the quality of the software?

A list of quality criteria for our project were formulated by our team members in a scrum meeting. Quality assurance activities were also discussed which were:

1. **Regular scrum meetings** where the risks which might affect the project were discussed and mitigated by the team members sitting together.
2. **code examination** is done in scrum meetings only while everyone is present as no one in our team is that good at coding. Necessary changes are done there itself or depending upon the situation they were done by the person responsible independently.
3. **Testing** is done to ensure the quality of the product; various types of testing are done such as:
    - *Manual testing* where expected behaviour is ensured by testing its code manually by testers.
    - *Integration testing* where testing is done to ensure that no bugs are occurred while integrating the features which were developed separately.
    - *User Acceptance testing* is done where testing is done to improve the time in which a user can understand operation of the application. Where any user tests the usability of the application and necessary changes are suggested if there were any.

3.5 To what extent do you think these processes, methods, and tools were effective in ensuring and/or improving the quality of the software you produced?

- Potential risks which might affect the project were identified and mitigated mostly during scrum meeting, by discussing among ourselves.
- Code examination by team members assessed properly working of the code. As we had less skilled members in coding, this helped a lot in improving features of our product to good extent.
- Integration testing made sure that tasks completed were all working as expected even after integration of features. User acceptance testing helped us in improving the usability of our application by considering the suggestions given by our friends, by changing the location or functionality of any feature.
- We made sure that list of all the quality criteria are met and rechecked them before final presentation of the product.

Question 4 - Project Team

4.1 In what ways do *you* think your team managed the project well? In your answer, provide specific examples, and explain why you think your team was good at these aspects.

- Team work was the main criteria which helped us to complete the project even when few team mates had very less development skills. Our team was very supportive and worked together as a team and ensured success of the project development. *For example*, team mates who were out of town and working remotely is a risk which might cause delay and difficulty while resolving issues occurred. Usually issues were resolved in status meetings with the help of all our team members which is not possible as he is away. In such case, we as a team supported and helped him by

conducting a status meeting and connecting him through skype and resolved issues by discussing changes to be made to ensure mitigation of issues.

- Project monitoring was done with the help of few tools; selection of tools was done by considering which tool might help us the most and accurately. *For example*, Burndown charts were used to know the time left for completion of project as it is work left vs time graph. After completion of every sprint, burn down charts were prepared for that sprint, deviation of actual line from idle line showed us if we are in correct phase or deviating from the project and necessary actions were taken to avoid the deviation.
- Every team member was ready to learn new concepts for this project, without any frustration or disinterest and strived their best to indulge themselves in developing our product. For example, not everyone from our team was good at developing web pages, for this project we had to learn html and CSS to participate in development process. Although development process was slow as we had to learn and do coding, because of our planning we could finish development of the product in time.
- All the suggestions which we got in status meetings were noted by some or the other team mate and were discussed in scrum meeting held after every status meeting and improved our project accordingly. For example, in the third status report where we have submitted our first burn down chart, it was not correct. Suggestions from the investor during the following status meet were discussed by us and corrected our burn down chart and made sure that we do it correctly in next status reports.

***All these shows efficiency and interest of our team members which lead to successful completion of our project. Our planning before the start of the project helped us to tackle risks occurred and maintain quality of the product we developed*.**

4.2 In what ways do *you* think your team could have managed the project better?

- Effort estimation could have been done better, as we lack the importance of effort estimation, it was neglected at first. Effort estimation on the tasks were done by using planning poker, and the estimates were very rough as we didn't have any previous experience working on such projects. They were very unrealistically estimated. In a status meeting we got a comment regarding our effort estimates then by discussing among ourselves we changed the estimates depending on the effort on completing similar tasks in our project.
- Our initial project plan was failed as we were a bit careless while presenting our plan, this lead to delay in starting our sprint 2 as we had to improve our project plan by submission date. As project monitoring, risk mitigation strategies were not included, justification for using scrum was not given, goals for quality criteria were not stated resulted in failure of our project plan. This was due to lack of identifying their importance during project planning stage, if only we were cautious, identified their importance and described them firstly, this rework would have been avoided.
- At first we assigned tasks to team members randomly, dependable tasks were assigned to two separate team mates, if one task had encountered a bug in it and could not be solved by the team mate responsible for it, as he had poor coding skills. dead lock is occurred, as the other dependable task cannot be completed without completion this task. Then that task was assigned to the other team mate who

completed both the dependable tasks or had to wait till he removes the bug. If at all we were cautious in assigning dependable tasks to one person, this waste of time due to deadlock would have been avoided.

- Tasks for executing "recent searches" from sprint 2 and "list of favourite searches" from sprint 3 were completed by a team member, but due to lack of good interface, those tasks were taken over by other team mate and were completed, but this resulted in late completion of these tasks and delay in completing next tasks as both tasks were completed in sprint 3 itself. This was due to waste of time as both the team members worked on same tasks. This could be avoided if we had followed the design perfectly without deviating from it.

Question 5: My role

5.1 How did *you* contribute to the success of your team in terms of (a) managing the project; and, (b) creating the software product?

- As a part of project management, I was responsible for monitoring our project. Tools used to monitor our project were stated in our project plan, have helped us in monitoring the project well.
- At the beginning of the project, I have participated in planning where the necessary features for our weather report application were sorted out by us. Features suggested by me were list of recent searches, list of favourites, etc. these features increased the usability of our product to great extent.
- I was the designer for this project and gave an architecture of our application "Wow-Weather". Though the design was good, we as a team members agreed to improvise it depending on the execution simplicity of any feature, as none of us were very good at coding. For example, few features like list of recent searches and list of favourites were developed differently from the design, this was due to time constraint, as creating drop down menus for those features take more time than showing them on left side of the screen. Other features were created as per design itself.
- To participate in development process, I have learnt html and worked on front end part on few tasks like settings window, adding appropriate images to weather report. I worked in pair with a team member (pair programming) on few other tasks like list of recent searches, about screen. As I was not good at coding only few tasks were assigned to me which resulted in my project failure, as I couldn't show any evidence on my learning period and I didn't work on many tasks.
- For user acceptance testing, I was responsible for collecting suggestions made by few of my friends, which were informed to my group and then depending on the discussions made by us considering other suggestions, few tasks were modified to maintain usability of the product.

5.2 Do *you* think the roles you had during the project were the most effective use of your skills and knowledge?

By working in this project, not only my skills were used effectively, they got even better. my theoretical knowledge was improved by facing real time experiences, which improved my knowledge in those areas. Examples of how my skills were used and what did I learnt from my project are stated below:

- My coding skills are very poor, all I had was theoretical knowledge, but for this project I learnt coding in html and did development part.

- Same with scrum methodology, I had very good theoretical knowledge on scrum methodology, by working on scrum in this project had helped me to get real time experience. Working as a team, participating in scrum meetings, monitoring our project, estimating effort all these were known only theoretically, but I lacked real time experiences which were obtained by working in this project.
- I played the role of designer in our project, where my knowledge on usability of the web page were effectively used, more over through user acceptance testing this knowledge got even better by considering other's point of view.

5.3 If you were to perform the same roles on a similar project in the future, what would you do differently?

- While assigning tasks to team members, dependencies among tasks will be analysed and are assigned accordingly. We faced a risk which was due to ignoring dependencies while assigning tasks to team members which was stated in 4.2 3rd point, assigning dependable tasks to one developer who has good skills or assigning task to be completed first to the one with good coding skill, would avoid dead locks in the project.
- Effort estimation was done vaguely as we didn't have much knowledge on developing any feature and how much time it takes to complete it. By working on this project I got to know how much time it takes to write some amount of code. So effort estimation will be done more accurately in my next project.
- As stated in 4.2 2nd point, our first project plan was rejected as we didn't mention few main criteria's, this was due to lack of knowing the importance of project monitoring and risk mitigation strategies. From the experience of this project, importance of these tasks is well known, so therefore it is clear that next time I can write better project plan without missing any important tasks.

5.4 What are the most important aspects of software project management that you have learnt by participating in this project?

My theoretical knowledge gained particle experience's in many aspects such as:

- Attending Scrum meetings, gathering requirements, disintegrating features into small executable tasks, effort estimation, identifying potential risks and their mitigation strategies, testing all were known only theoretically, by working in this project I gained real time experience of how these are to be managed and applied correctly.
- Working as a team, working with team mates with different nationalities, cultures, skills, producing our best by working together, all these experiences have occurred as we had team mates with different nationalities, skills, attitudes, working on this project made me learn how things work while working globally in any company. These experiences would surely help me in my future projects.
- Potential risks which might occur while working on a project and their effect on project were observed directly and how these risks can be avoided or mitigated is also learnt. Effect of risk, if a team member is working remotely can be communication problem especially while discussing about any issues faced by him is quiet difficult through conference call.