Contents

Introduction

As part of the module, each student was put in to a team of between 4-6 people. We were given the task of designing a stand-alone package to provide visitors with a multiple-choice quiz, to be used at engagement events like open days. The quiz had to contain 10 questions per topic and be easy to use. The engagement team had to be able to edit the quiz, be that edit questions, answers, topics, visitor information etc. while ensuring that no unauthorised persons are able to edit these aspects.

Team Organisation

Team Development Process Model (Bruce W. Tuckman, 1965)

Throughout the entirety of the module, team organisation was paramount to the success of completing the task. According to Bruce W. Tuckman (1965), each team will pass through 5 stages on their way to becoming a cohesive and productive team.

Forming

The first stage in the process is "forming", where teams come together for the first time. I believe that our team's experience definitely matched this stage as to begin with, the members of the group were very eager to get to work and were motivated to do the absolute best that they could to succeed in the task ahead. Myself and the others in the group began to form relationships with each other and I felt that I had been lucky in being put in to the group that I was. However, at this point no one was really confident enough to assume a leadership type role or take charge of the project, so we were reliant on Helen Phillips (the module lecturer at the time) to explain what was expected of each member of the team. This information was given to us through a "Code of Conduct" that we were to follow.

Storming

The next stage of Tuckman's' "Team Development Process Model" is "Storming". According to the model, this is where the team should have the least amount of effectiveness due to differences in how each feels the project should proceed and their unwillingness to accept group decisions. However, I find that this was not the case in our group and that this was the point in the project that each individual was willing to compromise with each other the most. The individuals in our group were

willing to put the team above themselves and were either unable to see or willing to look past the flaws in each other's characters. The group were happy to be democratic in the process of deciding on how the project should go forward, for example who should focus on what area of the project. One example of how this model did not apply to our group was that in this stage of the team's life cycle, there were no power struggles or competitions for attention. In our team, a de facto leader emerged and it did not seems like anyone had any issues with this. People were willing to accept Michael as the team lead and the project went ahead.

Norming

The next stage of Tuckman's model is "Norming". According to this model, this is where disagreements in the team begin to get resolved and the team builds up trust. The project starts to become more productive and members build up confidence and self-esteem. I partially agree that this stage could be applied to our team. The speed that the team could achieve outcomes increased. For example, each member of the team was delegated a task (use case diagrams, use case descriptions, functional requirements and non-functional requirements) for our first coursework hand-in. We set a deadline for when each piece was to be completed so that they could be curated and handed in. Since the team members were starting to gain confidence, one of the team members was able to speak up, expressing concern that he had other deadlines for that date and that he may not be able to complete the task to the best of his abilities. The team were willing to compromise and moved our set deadline to a new date to accommodate for this situation.

Performing

The next stage in the model is "Performing". This is where productivity is at its peak and there the team works together in a way where the team is more than just the sum of its parts. I believe that we our experience as a team definitely matches this model. I believe we hit this stage in the project around January/February time. It was at this time when we were working on the second coursework (Design and Planning & Risk Management) I felt as if the team worked at its finest. We were quick and effective in finishing tasks and resolving any disagreements. For example, our group disagreed with the seriousness of some of the points included in our risk assessment. However, we quickly worked through this disagreement through

discussions with people arguing their cases and the group democratically deciding on what they felt was best. This quick working through of disagreements meant that no productivity was lost and the group could keep working well. This meant that in this stage, the work we produced was the best that it could be. This was reflected in our marks where we achieved our highest score yet.

Adjourning

From here, the model states that the team could go 1 of 2 ways. Either the team resets to the "Forming" stage as group membership, leadership and purpose changes, or it declines into the "Adjourning" phase as the group becomes complacent. The dorming phase was not a part of Tuckerman's original model but was added by Tuckerman and Jenson (1977) later on. Based on my experience within the group, I feel that we went down the path of the "Adjourning" phase. As the project was coming to an end and we were given our final task of building a prototype for the quiz, our team effectiveness gradually declined. Members in the team - including myself - began to get complacent based on the work that we had completed previously. We still delegated tasks like previously, however at this point people were not as punctual or efficient as they were earlier in the project. Team members began to miss deadlines, team communication began to slip and, in our case, the team began to resent one member who we felt was not pulling his weight at all. There were attempts to communicate our frustrations, but they were not effective. It was at this point where the mindset of the team was not to do the best that we possibly could, but to take on the workloads of those not pulling their weight in order to simply finish the project and move on.

Barriers to teamwork

One of the barriers that we faced in working as a team was communication issues. Throughout the project but in-particularly towards the end, there were instances of communication issues. To start with, our team was supposed to contain 6 members, however one of the members in our team had dropped out of the course without informing neither the group or the university. This meant that we just had to continue, not knowing whether or not our group would expand, which would change the dynamic of the group. Particularly towards the end when it came to building the prototype, our team suffered from a lack of communication. For example, we would

cause merge conflicts that took up time to resolve and we would not know what was needed of each member. Our solution to the communication issue was to use various channels to make sure that we could be in talks with each other whenever was necessary. This included weekly meetings and a Facebook Messenger group where we could communicate what was accomplished and what needed to be done in what timeframes. We would also refer to the Gantt chart in these meetings so that each member of the team could see what needed to be done and what time we had to do it. This initially worked with every team member attending the meetings and talking frequently in our Facebook chat. However, it started to slip, with some members making excuses as to why they couldn't attend a team meeting and the Facebook chat, while still being used, was used less and less frequently. Even with this slip, we still communicated better than before the meetings and the chat existed, so I would say our solution was partially successful.

Managing the Software development process

Techniques used in software development

Throughout the module, I was introduced to many of the different techniques that help development teams to manage the development of a project. The technique that I found most helpful for producing quality software was the "Gantt chart". I believe that this was the most beneficial technique to development as it gives the team an outline of the entire project. A good Gantt chart will give the full scope of the project from the very beginning to the very end. It was useful as it enabled the team and I to look back and see just where we were in the development process. In the Gantt chart, there are dates that you can refer to in order to see how much time that each aspect of the project is estimated to take and the team can prioritize certain aspects in order to meet the planned durations specified by the Gantt chart. If we had forgotten anything or were unsure about the next steps that we needed to take in the project, we could simply refer to the Gantt chart and we would know exactly where we were and what we needed to do, which I personally found to be a huge help in development.

However, there were some techniques that I found were of little use during the development process. The technique that I found was of least use was the "Risk

analysis" document. The risk analysis was intended to outline any potential risks and challenges that could disrupt the development process and outline strategies to mitigate this disruption. However, I found that the team and I after making it, barely used it. The team preferred to go through the development process and tackle any challenges on the fly, rather than referring to a set strategy that was outlined in the Risk analysis document. It also seemed as though some of the risks that we outlined in the document could be overcome through common sense solutions rather than something we had to specifically plan for. For example, one of the risks that the team outlined in the document was that we could "Fail to submit the project before the deadline". This, to me, seemed like a redundant risk as I felt that there was no possible way that the team could simply forget to submit the project. If we were not fully finished and the deadline was soon, we would just submit what we had done up to that point. It was a document that I feel we made simply because we needed to make it for the hand-in of the coursework, rather than something that we would use over the course of development.

Project management

In terms of project management, I believe that my team and I worked well to make sure that everyone knew what needed to be done. We delegated effectively and used the various documents such as the Gantt chart and use case descriptions to make sure that we knew exactly what the project needed and what timeframes we would require to achieve these needs.

Risk management

Our use of risk management strategy was underwhelming. While we did make up documents outlining the potential risks and the strategies we would use to overcome these risks, we did not really use any of them and the team would deal with any challenges on the fly. Instead of referring to the documents we had produced, we would offer solutions as a group and then vote on which one to go ahead with. We did this as we felt that at the point of creating the Risk analysis document, we were not able to predict all of the risks and challenges that we might face, so we would just deal with challenges as they appeared instead.

Version control

The teams use of version control was ok but could have been better. In the prototyping stage of the quiz, we would use GitLab to push major changes, but not necessarily smaller tweaks and modifications, as some of the group felt that these small changes were not worth the effort of re-adding and pushing. This lead to some merge conflicts where two people had changed things in the same file without communicating it to the group. Since this was in the later stages of the team's life cycle, communication was a problem as was outlined earlier in the report. If we had continued to communicate with each other like we had been doing earlier in the project, I believe there would be no problems of this sort. So, while we did use things like Gitlab for version control, the communication between the team to tell of new version pushes, what people were working on etc meant that we were less effective and had to spend more time resolving merge conflicts.

Quality criteria

According to McCall's Quality Model (1977), there are various factors that a development team should consider in order to develop a high quality program, system etc.

Flexibility

One of the parts of the final product that I was involved in was constructing a way of of loading and saving questions banks in to the quiz. One of the quality criteria that I had to take in to account was "Flexibility". I took flexibility in to account by making sure that the engagement team could quickly and easily switch between question banks existing banks or make new ones, depending on what type of theme they wanted to run with the quiz. This modularity meant that the quiz could be used for any themed event that the engagement team want to run. The questions are stored in csv files with an easy to follow format, so it is easy for the engagement team to either make new files for new themes or edit existing ones. This enhances the quality of the final product as it means that the engagement team are able to adapt the quiz to whatever event they need to run.

Integrity

Another quality factor that I had to take in to account when building the question bank functionality was "Integrity". Integrity means making sure that the program is protected from unauthorised access. I needed to make sure that the tools that the engagement team had in order to edit questions in the quiz could not be accessed by unauthorised persons. I did this by making sure that the engagement team had to use the login feature first in order to access the tools needed to change the questions. This improved the final product as it meant that it was protected from vandalism, as only the engagement team had access to these tools necessary to edit the questions.

Usability

Another part of the final product that I took part in building was the ability to store information on the schools that were visiting the event. The engagement team were able to input what schools were attending the event so that they could collect statistics on how well each pupil from each school did on the test. A quality characteristic that I had to consider when building this functionality was "Usability". Usability takes in to account how easy it is to learn, operate, prepare input and interpret output of a program. In this case, I took Usability in to account by making sure that the system I built was well documented and easy to follow. There are on screen instructions during the process of adding and editing the schools. These instructions give the exact formatting of the input needed to add schools and if the user inputs something that doesn't match the format, the program will tell the engagement team and allow them to amend the error. This enhances the quality of the final product as it means that the program requires little effort to learn and operate, meaning the engagement team can spend as little amount of time as possible actually interacting with the program for it to give the required output, e.g. themed quiz.

Efficiency

One other quality factor that I had to consider when building the school information storage functionality was "Efficiency". I wanted to make sure that the functionality was achieved while also making sure the program still ran fast and smooth, while taking up as little space as possible. I took this in to account by making sure the

code was as concise as I believed it could be while still maintaining functionality. This benefitted the final product as it meant that the program could be run on practically any hardware while still running perfectly fine, keeping the costs needed to run an event low.

References

- McCall, J. A., Richards, P. K., and Walters, G. F., "Factors in Software Quality", Nat'l Tech.Information Service, no. Vol. 1, 2 and 3, 1977.
- Bruce W. Tuckman, “Development Sequence in Small Groups”, Psychological Bulletin. 1965.
- Bruce W. Tuckman, Mary Ann Jensen, “Stages of Small-Group Development Revisited”, Group & Organization Management, Vol 2, Issue 4, pp. 419 – 427, 1977