Table of Contents

Introduction

The project is about creating a prototype for a standalone platform quiz game which will be used during schoolings events to test the ability of students for the engagement team. It is a simple prototype that will be easy to use my students and also managed by a member of the engagement team without a programming experience. As part of the requirement it is stated that the engagement team must be able to add, remove and edit both the questions and information of the schools. In other to carry out the project we were placed in a team of 6 members to work together to create the prototype using the engagement team specification.

Team organisation

A team is a group of people with a set of complementary skills required to complete a task or project. In other to carry out this project we worked as a team to accomplish the purpose of the projects. According to the business journal "group" is a crew of people who arrange their individual endeavours, where as a "team" is a crew of individuals who share a typical team goal and various challenging goals. Individuals from the teams are commonly dedicated to the objectives and to each other. This common duty likewise makes joint responsibility which makes a solid bond and a solid inspiration to perform (Sisson, 2013).

To evaluate my team i am using the Bruce Tuckman's stages of team development.

Tuckman's stages of team development

Forming

This is when we were first put together as a group. We started by introducing ourselves and doing a little ice breaker by disclosing 3 things about ourselves that includes our hobbies and things that we dislike. We also discussed the skills and knowledge that we have that might be of use in the project. At this stage we all didn't know how complex the project will be and what is required.

Although it is required at this point to have issue with authority as there will be conflict on who to be the team leader or not. We only nominated someone to be the leader but decided that all decision will be taken together, and no one is superior than the other.

Storming

After the introduction we all came together to draft a ground rule and constitution that will govern us as a team and make sure we all stick to it. We also at this point decided on our means of communications and what platform we can use to work together to make it easier for everyone, and the days we will meet outside of the normal tutorial hours. We planned on using Facebook, discord, WhatsApp and emails as means of communication, and for workspaces we choose OneDrive and GitLab. Although we choose some of the platforms, but we didn't use most of them, we only use the most effective one which is WhatsApp, OneDrive and GitLab. We also unanimously agreed on the days of our meeting and what software we are going to use.

At this point we know what is expected of us in the project, and what roles are needed so we all shared the roles based on our skills and ability using the Belbin's team roles. Although it was an unproductive phase, we didn't have issue of authority as we have already decided on how decision will be made. We also didn't have problems with individual task as we sit together to do the work. The only issue was that not everyone was available at every meeting.

Norming

This is the third stage, at this stage we have starting planning on how to carry out the projects and what we need to do to complete the project. After working together as a team for months, we have the same system of meeting together to do the task together. We had different views on how we think the design of the interface should look like and what is expected of us by the client. We didn't share the tasks between ourselves except the development process as it might take a while to get everything done if we sit to do them together.

At this stage the only issue we have as a team was that not everyone is available at all meetings. To involve everyone in the decision making and content of the project, we ask for opinions on the group chat from those who are not there. With this we were able to avoid the issue of not carrying everyone along.

Performing

According to Tuckman the team at this stage is more deliberately mindful; the team knows plainly why it is doing what it is doing. The team has a common vision and can withstand it with no obstruction or support from the leader (Businessballs, 2017). Applying this to the team we didn't have any issue as a team getting to this point. We all know what is expected of us to be done, every member has to be in the meeting for the development process. We carried out different research and feedback to the team and carry on with the development of the program.

Using the Belbin theory, I will say that we worked as a team and not as a group because we did most of the task together. We did the planning, the design and the decisions we make were together. This was a very productive stage for us, as we had to work on the projects all the time to meet up with the deadline. I was afraid at some point that we might not meet up the deadline due to the holidays, but we were able to pull though as a team when we all came back from the holidays to meet up with the deadline. We did the rigorous testing of the different parts of the software, to make sure it was up to the expected quality.

The software development process

Planning

In other to carry out any project planning is very important, this is why we decided to create an activity plan table within the time frame needed to complete the project. This is to serve as a guide and let us know if we are ahead of our schedule or behind. The planning has 3stages

the gathering and understanding of the user requirement from the client, planning the programs which involves the backend and testing the program after its been created.

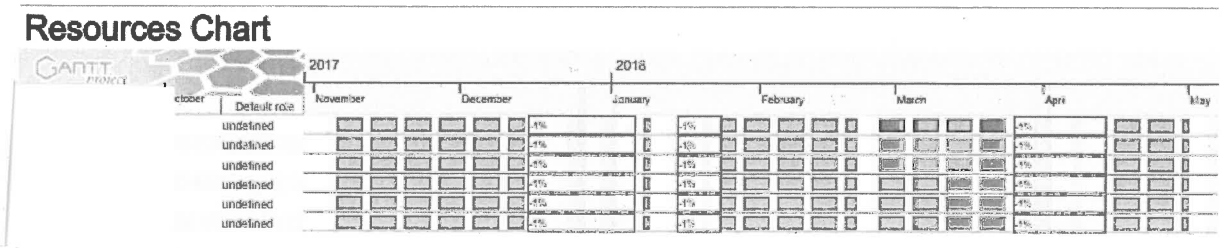
Sample of the Activity Plan

DATE	ACTIVITY	DURATION IN DAYS
30th Oct 2017	Drafting of team contract and assigning roles	1
	Requirement	
6th Nov 2017	Meeting with the leader of the engagement team to discuss user requirement	1
13th Nov 2017	Drafting of functional requirements	2
20th Nov 2017	Drafting of nonfunctional requirements	2
27th Nov 2017	Validate the requirements with the leader of the engagement team	1
6th Dec 2017	Create a use case description of the functional and nonfunctional requirement	4
11th Dec 2017	Create UML diagram of the user requirement	2
18th Dec 2017	Break	1
25th Dec 2017	Break	
1st Jan 2018	Break	
8th Jan 2018	Meeting with engagement team	1
15th Jan 2018	Break	
	Planning	
24th Jan 2018	Produce a use case description for the requirement	2
29th Jan 2018	Produce a class diagram	4
5th Feb 2018	Review of the class diagram and use case	1
12th Feb 2018	Produce an action plan using Gantt chart for the design and development of the system	2
19th Feb 2018	Identify and produce a risk assessment in planning and development	2
	Design and Implementation	
26th Feb 2018	Design a prototype of the interface of the quiz	4
5th Mar 2018	Meet with the leader of the engagement team to review the design, background and font colour	1
	Creating the backend	
12th Mar 2018	Start coding using java	3
15th Mar 2018	Creating table on SQL	1
16th Mar 2018	Linking the java program to the table on SQL	1
	Creating the frontend	
19th Mar 2018	Creating navigation buttons	1
19th Mar 2018	Creating background	1
20th Mar 2018	Creating and adding dropdown menu	1
21st Mar 2018	Creating front page and questions layout	1
22nd Mar 2018	Creating summary page and compiling the program	1
23rd Mar 2018	Adding data to the backend	1
26th Mar 2018	Break	
9th April 2018	Break	
16th April 2018	Testing the quiz with the use case description	2
23rd April 2018	Making final adjustment based on the result of the test	2

30th April 2018	Meeting with the Engagement team to hand over the final Quiz program	1

The table above is the sample of the activity plan we created for the project, the plan covered what we are supposed to do over the time of the project, showing the date and duration each task would take to finish. However, a plan is only assumption of what we expect to do and when to finish. We tried to follow the plan closely as possible but there were unexpected surprises that came up.

Sample Team Management



We used the Gantt chart to manage the team work, we made sure each member of the team contributed to every aspect of the project and in cases where it is necessary that we must split to do a task we split into two or three and do them in groups.

Risk Analysis

We conducted a risk analysis of the project and used the risk map below to measure the level of risk, if it is high, low or moderate. We rated situations that are more likely to affect our projects in a bad way as high such as team members struggling to do the work, which is one of the reason we decided to work as a team rather than individuals. In addition, we rated situations such as illness of team member as moderate, because this may affect the project if it's an individual task but will not have such an impact if it was done as a group task. Also, we rated situations such as corrupted files, inactive member, mismanagement of time and team conflict as low because if such situations arise we will be able to handle them without affecting the project itself. We all have copies of the files that we are working on, in a case where the file gets corrupts it can be easily replaced with a new copy that is not corrupt. Other issues that may arise has been discussed in the team contract.

Sample of Risk Map

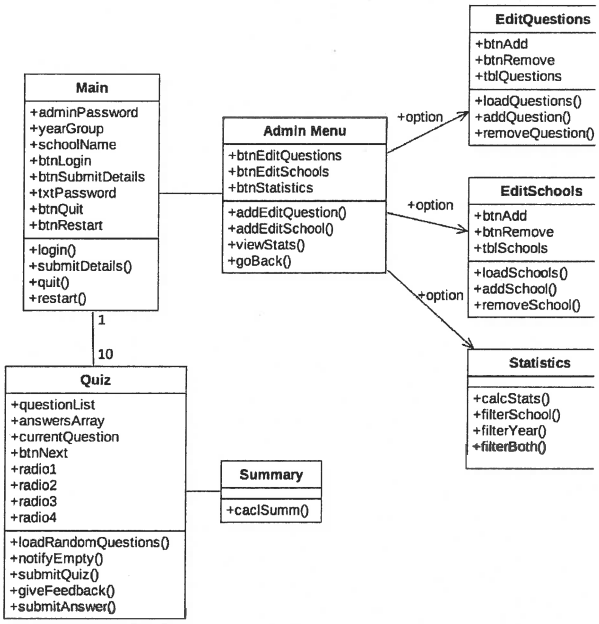
Risk Map		**Likelihood of occurrence**		
		High	**Medium**	**Low**
Potential Scale of Impact	**Large**			1 2 6
	Moderate	5	3	4
	Small			

Design

The designing of the interface was done in a way that is simple and easy to understand by the users such as the students and the engagement team. The design involved using a dropdown menu for the simple selection of schools and year group. The users are also given a choice to select from a range of topics. When they start the quiz, they get instant feedback on answering any question and have a bonus question at the end of each quiz. Both students and engagement will be able to quit or restart the quiz at any point. The background is white and the font colour black, which was consistent throughout the quiz. Its an easy interface to enable users see the text and easy use of the software.

As part of security, the admin section of the software is protected with a password that only members of engagement team or anyone that is in charge will be able to access it. This was put in place to restrict student access to the backend to change questions or try to access the code. The UML diagram was used to create the design.

Sample of UML Design



Developing & Implementation

The development of the quiz was in java programming language using NetBeans. We planned on making it a text base quiz but then changed the plan when we realize we can make a qui quiz with NetBeans. We failed in our first two attempt to start the development, after getting the hang of the interface we decided to continue with NetBeans. In addition, extra work was done during the holidays to enable the team to finish the work before the deadline. Gitlab was used as the platform for sharing the codes as we all can work individually and commit it to the GitLab for others to see the changes. After the holidays we met as a group to finish the remaining part of the coursework.

The UML diagram was of great help and used when writing the code. This is because it was

used to determine the connections between the different frames in the quiz such as inheritance, aggregation, composition, dependencies and generalization. In addition, the use case description was also used to determine the alternatives of the quiz flow. While creating we had to use inheritance and polymorphism to make it easy to add more features to the software.

Testing

The testing of the quiz was done after the development of the program. We used the use case description of the functions to check if we met the user requirements of the software. As individuals we conducted our own testing of the software to make sure it meets the user requirement and is of good quality, and as a group we conducted a few as well.

Quality criteria

Security

The picture below shows the code for the security of the software. The password is what prevernts other users from getting access to edit or remove questions from the question bank or edit the school data.

```
private void txtPasswordActionPerformed(java.awt.event.ActionEvent evt) {
    if (txtPassword.getText().trim().toLowerCase().equals("teamG")) {
        AdminMenu menu = new AdminMenu();
        menu.setLocationRelativeTo(null);
        menu.setVisible(true);
        dispose();
    } else {
        System.out.print("WRONG");
        JOptionPane optionPane = new JOptionPane("Please Enter The Correct Password", JOptionPane.WARNING_MESSAGE);
        JDialog dialog = optionPane.createDialog("Password Incorrect");
        dialog.setAlwaysOnTop(true);
        dialog.setVisible(true);
    }
}
```

- Usability- the security is easy to use as long as they know the password.
- Realibility- it is reliable because without the password they cant get access to the admin interface.
- Intergrity – it prevents unauthorised access to the admin interface.
- Maintainability – its easy to maintain as they can choose to change the password when they want by adding to that portion of the code.

The project

Usability & Flexibility

The software is easy to use, and users do not require a knowledge of programming to be able to use the software. The management team can easily add or remove schools and classes in the admin section of the program with ease. They can only use the buttons that are provided to either remove or add the schools. This also is the same for the questions as they can easily remove and add questions to the question bank. The software is programmed to only show 10questions for each player, with this the engagement team do no need to code or change anything to the code but instead use the GUi to add and remove questions that are stored in a separate CSV file when entered. The font style and colour are simple to enable people see and read the questions and

information there. There was nothing fancy except the home screen, this is to make is easy on everyone that will be using the software such as the students and members of the engagement team.

Reliability

To check the reliability of a software entails testing the software ability to function and produce the expected result, the environmental conditions and duration. We tested the software separately to ensure that it doesn't break and give errors, but it didn't give error after series of test. Before the final testing was made, we found bugs that was fixed before handing it over to the engagement team. This is to say that there is certainty that the software we made is reliable and can withstand a long run time on any computer that it will be used.

Maintainability & Integrity

The software is easy to maintain, as the engagement team do not need to make changes to the code. All they need to do is to use the interface during events. The security added makes it easy for only members of the engagement team to add or edit the backend of the interface.

Conclusion

In conclusion working as a team has been more challenging, fun and educating at the same time. This is my first time working as a team and not as a group and it has proven to be more effective than when working as a group. Splitting the work to do them individually will affect the projects if that individual get sick or loose their work and refuse to start again, but doing them as a team or little groups in the teams makes it easy as someone will have a copy of the lost work or the other people doing the task can complete it in the absent of the sick individual.

Planning a project is good for both team management and time management. This is because it will enable the team to focus on meeting the targeted deadline and making sure that the project is delivered on time. The risk analysis we carried out for the project was good as it helps to overcome those foreseen issues, but not so much of the unforeseen. One thing we didn't do as a team that I thought would have been helpful was to carry out a critical path analysis of the project, this would have helped to know the major milestones in the creation of the software.

Generally, I will say the quality of the software is good for a prototype and can be duplicated due to its reliability to create a more efficient and better software that will be used across different platforms such as window, mac or Linux computers that are compatible with NetBeans or have a java programming software installed.

References

Businessballs, 2017. *Tuckman forming storming norming performing model.* [Online]
Available at: https://www.businessballs.com/team-management/tuckman-forming-storming-norming-performing-model-234/
[Accessed 04 05 2018].

Sisson, J., 2013. *The difference between a group and a team.* [Online]
Available at: https://www.bizjournals.com/bizjournals/how-to/growth-strategies/2013/06/the-difference-between-a-group-and-a.html
[Accessed 02 05 2018].