Approach to Change Management

After assessment 2, we had to decide which other groups project to take on and work on further. We decided this as a group after going away and evaluating other group's projects. We each chose some projects that we liked the look of and brought our ideas to a meeting where we discussed each project's advantages and disadvantages. For us, some of the most important aspects of the project we chose was the code readability, the quality of their documentation and how confident we felt in completing the specified requirements for the project.

In our meeting to decide which group to take on we shortlisted 3 projects. We evaluated the code of these in a more in depth approach and decided that "Space Key Projects" would be the best project for us to take on, due to their excellently commented code, well written documentation and an architecture similar to our own, making the transition of moving to the project easier. After this, we spent some time working through the group's code to fully understand the project we had taken on.

After looking at Somverville's Software Engineering [1] we decided to simplify down a version of the "change request form. We have called this "Change Log" https://marcelmiro.github.io/SEPRSpaceKey/Assessment3/Test%20Plan%20and%20Test%20Results/Change%20Log.pdf . This allows us to accurately keep track of the changes we are making to the project. In this change log each row would represent a change, we include an "Requirement ID" which is a reference of the requirement(s) that this change is affecting/implementing. When a change has been implemented and tested we would mark it down as completed in the change log as green, if a change has been partially implemented we would mark it down as orange/yellow, if a change has not been implemented yet it would be red.

In the change log we also include a description of the change, the member of the team assigned to deal with the implementation of the change, the impacted classes/files and and testing resources. This will give everyone in the group a good idea of what changes have been made, by who and what effect it has had on the project. For the testing, we will run all tests after each change has been made, including new tests for the new change. This will allow us to make sure that everything that nothing has been broken by making the changes described.

Changes to the project's documentation will be done by the group members assigned to the different documents. These changes will be identified by highlighting the changes to the original documentation, this will allow anyone looking at the documents to quickly identify the important areas of change. The group members in charge of each deliverable are: Ben and Amy for the Change Report, Jay and Fiaaz for the implementation report, Rian, Marcel and Ahmed for the implementation and Marcel for the website

Changes to Testing Report

Overall the previous teams testing report was well done. The decisions on how to test their code and also the creation of the tests themselves seem well justified and work well to prove that the overall game works and passes the requirements set by the team. Our team has decided to make minimal changes to the way that the testing has been completed and will only add additional tests based on the additions we have put in the game.

- The implementation team also completed white box testing as they wrote the code. To continually check the functionality of the game and tweak its difficulty.
- Junit tests created by the original team will continue to be used to show that the main functionality of the game works.
- Black box testing will be used to show the new code also behaves in the way that it is expected to for the user.
- Acceptance testing will be used for the new implementation to show that the new code meets the additional requirements laid for assessment 3.

The layout of each of the test result tables is concise and easy to follow so minimal changes needed to be made to the report structure. Only one minor change was added: relating the figure values in black box testing to their corresponding test.

Once the project was taken over by the group the testing team initially needed to check that all the requirements had been tested by the team. This was difficult to work out from just the test suite and requirement table alone so we decided to create a traceability matrix https://marcelmiro.github.io/SEPRSpaceKey/Assessment3/Test%20Plan%20and%20Test%20Results/Traceability%20Matrix.xlsx to visualise what the previous team had done and to make sure that we would test all of the additional requirements created by our team as well. The initial traceability matrix showed the previous team had good coverage of their tests and no additional ones needed to be made.

However, when it came to actually running the old Junit tests because of the changes we had made to the code they would not run. This is because we added functionality to the old classes which meant that the old tests would not run. This resulted in us rewriting the already existing junit tests so that they would run. The Junit test code can be found here: https://marcelmiro.github.io/SEPRSpaceKey/Assessment3/a3page.html in the drop down menu labled “JUnit Test Code”.
We did not add any additional Junit tests as the majority of the new implementation falls under visual changes which can be tested with Black box testing. This is because very few methods have been added but have just been changed and used in additional locations in the code.

When it came to writing our own tests we first had to create the requirements to meet the additional demands of assessment 3. The new requirements can be found here https://marcelmiro.github.io/SEPRSpaceKey/Assessment3/Test%20Plan%20and%20Test%20Results/Requirements.pdf From this we could then create the additional tests (which are

black box and acceptance tests) that are needed to test all of the implementation that we added to the game. We then updated the traceability matrix to show that all of the new requirements have been adequately tested. The updated traceability matrix can be found here
https://marcelmiro.github.io/SEPRSpaceKey/Assessment3/Test%20Plan%20and%20Test%20Results/Traceability%20Matrix.xlsx
A file containing all of our testing can be found
https://marcelmiro.github.io/SEPRSpaceKey/Assessment3/Test%20Plan%20and%20Test%20Results/Tests%20Document.pdf . This includes details of the unit tests, black/white box tests and anything else.

Changes to Methods and Plans

Several changes have been made to the method and planning section to incorporate the previous techniques used by our team. However, we have decided to keep some of the methods the same as they were either identical or are good ideas which would be useful to our team for this project.
For the tools overview:

- We won’t be using Jira for project management so this has been removed from the document. This is because no one on the team has used the software before and it would not be a good use of our time to learn how to use the application. Our current method for project management under SCRUM has worked well so far. We create minutes after each scrum meeting for the following sprint and our arranged deadlines provide a good time frame for when to meet next and what needs to be achieved in each sprint.
- We will not use slack as our form of communication as our facebook messenger group chat is already well established and team members have agreed that it is more useful on messenger as they may forget to check slack so regularly.
- We have also added discord to the tools overview as we have found this useful for voice chats when group members could not meet in person.
- The implementation team will continue to use intellij as their java IDE as it can be connected to the GitHub repository directly instead of with Eclipse.
- For the architecture, the group has decided to change from Plantext which we used previously to lucid chart which is what all of the architecture is now on. This is because the team likes how the diagrams look and it is easy to learn how to use the software.

For team organisation we have decided to take on their informal approach as this is similar to what we are doing. As for the team roles we have decided to take on their roles with a few minor adaptations and reassign some of the team members to better fit this assignment.

- We have split secretary and reviewer into two roles.
- We have removed UI and Graphics designer as the majority of the graphics have been drawn and the main UI has been completed.
- We have added the HR role to the general risk manager.

The roles we have assigned to each member of the group have been added to the document and were selected based on the previous roles from the last assignment and also the tasks which team members have been given.

We will use their Ganntt chart for this assessment as it is well structured, easy to read and provides an achievable time frame for tasks to be finished in. There have been some minor changes to the chart; we have reassigned each of the project tasks to the correct team members from our team and we have removed “update jira backlog” as we have decided not to use Jira. The changes to the method and plan can be found here https://marcelmiro.github.io/SEPRSpaceKey/Assessment3/Test%20Plan%20and%20Test%20Results/Methods,%20Plans%20update.pdf

Bibliography

[1] I. Somerville, *Software Engineering*, Pearson, 10th ed., 2016