Change Report

Stakeholders: Richard Paige, University of York Communications Office
Team: Barney Morgan, Cameron Smith, Harry Berge, Jake Phillips, Matthew Wilkie, Rob Weddell

Change Management

Upon researching it was found that the most appropriate document in regards to change management was the IEEE 828 Standard for Configuration Management in System and Software Engineering [1]. Within this document, it states that it 'establishes and protects the integrity of a product or product component throughout its lifespan'. This was especially important to both establish that the previous team had worked to this standard, and to ensure that the current team continued to follow this standard.

Due to the fact the previous team had developed their project with an agile method in mind, it was relatively easy for the team to continue this project by adopting the roles and methodology of the previous team [2] [3]. However, it was found that the previous team's code was rather unclear and not very modular which resulted in many roadblocks during development. Consequently, research was done into the best solution to combat this struggle [4][5][6]. It was found that rewriting was often more time efficient for smaller projects, whereas with larger projects it was better to refactor the code. Thus, as the project is still relatively small, it was decided that the best approach would be to rewrite the code by reimplementing one feature at a time into a completely new project inspired by the previous one. This meant that the functionality was almost identical to the previous project, yet the implementation was more efficient and easier to work with.

An unintentional side effect of this decision was the loss of certain, smaller and insignificant, features in the process of rewriting. In an attempt to negate the impact of this, the team ensured they took into consideration the requirements table [7] left by the previous team. This allowed the team to track both newly and previously implemented features back to the requirements and ensured that all core functionalities remained present throughout the development cycle.

A change management log [8] was created in order to officially track any changes made and the impact of the proposed change to the overall project. This document takes inspiration from the proposed 'change request form' in the IEEE documentation. However, due to the small nature of the project, it was deemed that a much more informal approach could be taken. The team adopted the following criteria for the document: an 'owner' for each change request, the 'section' of the code or documentation it affects, the 'description' of the change, the 'impact' of the change to other areas of code, the 'date requested' and completed, and the current 'status'. This log, alongside the version control features of Github and Google Drive, enabled the team to effectively manage any changes to the code and ensured that everything was tracked.

Alongside changes to the code, the team was also required to update some aspects of the documentation such as the various testing documents. These changes were mainly due to the fact that so much of the code had to be rewritten that many of the previous tests simply did not work anymore. However, it was also partly due to the fact that the team felt the previous group did not fully cover all aspects in their previous testing and thus more detail was required.

Key:
The previous team's changes to their documentation during the second assessment are highlighted in blue
Changes to the previous team's documentation during the third assessment by us are highlighted in yellow

Justification of Changes

a. Testing Report

As the main structure of the code was changed to a less script-like implementation, most of the tests have also been changed to reflect this. A new set of unit tests have been created to effectively check whether the base functions continue to work as expected while the development of the game continues. Furthermore, as with the group's previous assessment, it was chosen to use continuous integration through TravisCI [9] in order to check the integrity of each commit made to the project. Specifics of the new unit tests can be found in the updated test document [10]. Peer testing was introduced as the other group did not use this, this can also be seen in the updated document to see how we used it.

b. Methods and Planning

Due to the fact that both the team's previous work and the chosen team's project followed an agile-method, taking over their workflow was relatively seamless. In particular, they followed a Scrum approach which meant that all of the previous project roles could smoothly be transferred into the new project. This meant that essentially all of the team organisation could remain unchanged. Despite the similarities between both approaches to team organisation, their assigned roles covered different aspects. It was decided to simply adopt the chosen team's predefined roles as opposed to overwriting them with new ones. This made it much easier to continue their work and was made easy due to the transferable nature of the roles assigned in the previous assessment.

In addition to this, the chosen project shared the same applications to complete their work: namely LibGDX [11] and GitHub [12]. This was one of the main influencing factors in the team's decision to take over this project as it ensured that the team had no issues transitioning over to the new framework as all the tools remained the same as previously comfortable with. This ensured that minimal time was spent getting to grips with new software and enabled the team to maximise the time spent on the project.

A few changes to the documentation were needed in order to reflect the chosen methods and planning of the team. It was found that the previous team did not plan this iteration of development with much detail - possibly because they were uncertain as to the requirements to complete this part of the project at the time of creating their plan. However, the main issue resided in the fact that both teams had a very different (yet equally viable) workflow and approach to the tasks. Because of this, it was deemed that a few additions and alterations were needed to the previous team's plan for assessment three and four. The Gantt charts inherited from the previous team demonstrated a waterfall method of working, whilst the plan stated employing an agile working method. This confliction also contributed to the redesign of their Gantt charts. As a result, it was decided to use our previous Gantt charts [13][14] for the planning of this assessment as they accurately represent our current agile workflow. As a result of this change, the plan [15] has been updated to mention the software used to create the Gantt charts, ProjectLibre [16].

c. Requirements

Whilst not necessary, it was found that ever so slightly updating the previous team's requirement documentation enabled the team to work more effectively to create a better product. The team was conscious that making large changes to the requirements would result in the project failing [17] as they were not given the task of changing the project, but simply continuing from where it was left off. As such, any changes to the requirements are simple alterations to ensure clarity (such as removing alternative implementation choices) or by removing redundant requirements.

d. Risk Assessment

Similar to the requirements, it was deemed essential not to make large alterations to the previous teams risk register. However, some small changes such as adding ownership to each risk were made to make the workload slightly easier for this assessment.

References

[1] IEEE Standard for Configuration Management in Systems and Software Engineering. [Online] Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6170935 [Accessed 15 Nov. 2018]
[2] Applying Configuration Management to Agile Teams. [Online] Available: https://www.cmcrossroads.com/article/applying-configuration-management-agile-teams [Accessed 15 Nov. 2018]
[3] Agile configuration management for large organizations. [Online]. Available: https://www.ibm.com/developerworks/rational/library/mar07/schuh/index.html [Accessed 16 Nov. 2018]
[4] Refactoring vs. Rewrite [Online]. Available: https://www.targetprocess.com/blog/refactoring-vs-rewrite/ [Accessed 16 Nov. 2018]
[5] Rewrites vs. refactoring: 17 essential reads for developers [Online]. Available: https://techbeacon.com/app-dev-testing/rewrites-vs-refactoring-17-essential-reads-developers [Accessed 15 Nov. 2018]
[6] The Rewrite vs Refactor Debate: 8 Things You Need to Know [Online]. Available: https://dev.to/bosepchuk/the-rewrite-vs-refactor-debate-8-things-you-need-to-know-2hi4 [Accessed 17 Nov. 2018]
[7] Element of SEPRise!, Updated Requirements [Online]. Available: https://sepr4.github.io/web/submission/assessment3/updated/Req3.pdf [Accessed 2 Feb. 2019]
[8] Element of SEPRise!, Change Management Log [Online]. Available: https://sepr4.github.io/web/submission/assessment3/ChangeLog3.pdf [Accessed 2 Feb. 2019]
[9] TravisCI Website [Online]. Available: https://travis-ci.org/ [Accessed 3 Nov. 2018]
[10] Element of SEPRise!, Updated Tests [Online]. Available: https://sepr4.github.io/web/submission/assessment3/updated/testing/Test3.pdf [Accessed 2 Feb. 2019]
[11] LibGDX Website [Online]. Available: https://libgdx.info/ [Accessed 5 Oct. 2018]
[12] GitHub Website [Online]. Available: https://github.com/ [Accessed 5 Oct. 2018]
[13] Element of SEPRise!, Assessment 3 Gantt Chart [Online] Available: https://sepr4.github.io/web/submission/assessment2/updated/gantt/Ass3.png [Accessed 2 Dec. 2018]
[14] Element of SEPRise!, Assessment 4 Gantt Chart [Online] Available: https://sepr4.github.io/web/submission/assessment2/updated/gantt/Ass4.png [Accessed 2 Dec. 2018]
[14] Element of SEPRise!, Updated Plan [Online]. Available: https://sepr4.github.io/web/submission/assessment3/updated/Plan3.pdf [Accessed 2 Feb. 2019]
[16] ProjectLibre Website [Online]. Available: https://www.projectlibre.com/ [Accessed 5 Oct. 2018]
[17] Best practices for project handover in middle-size organisations [Online]. Available: https://www.theseus.fi/bitstream/handle/10024/47111/Laine_Markus.pdf?sequence=1 [Accessed 14 Feb. 2019]