

Change Report

Change management

The first thing we did as a team was to have a good look at the code and deliverables we had inherited from Team Farce. We wanted to ensure we were familiar with the documentation and code, so we could make informed decisions on what changes we needed to make.

We were aware that we would need to mainly do two types of change:

- **Corrective changes:** We discovered some faults and errors within the codebase which we tried to fix.
- **Additive changes:** To complete Assessment 3 we needed to add a number of features in order to complete the game to the requirements specification.

We also needed to be aware of two other forms of change we may need:

- **Deletive changes:** We discovered some features of the other teams reports and code that did not work or were buggy. This especially caused problems when trying to use their methods and classes to implement new features.
- **Perfective changes:** We found that the games architecture was, in some places, badly organised and corrected this with extensive refactoring in order to make working with the code easier for ourselves and potentially others. This work is described in the implementation report, along with other code changes.

As a team we evaluated the advantages and disadvantages of the code and documents, then we used what we had learned to help decide which documents to work with and what changes needed to be made. We decided to keep all of our documents, and adapt them to suit the new project as we felt this was achievable as we were already very familiar with our documentation. This choice of documentation also made our lives easier when it came to traceability.

We also looked at the code and decided which of the change types mentioned above were required for each part. We analysed our requirements and looked for any changes that needed to be made, including adding in requirements related to the features that were already implemented in the game that we inherited. We moved on to implementation, where some parts of the code were kept the same, some parts were changed slightly, and some parts of the code were completely changed.

After updating our requirements, we began development on the changes and additions that needed to be made. We tried to ensure traceability of requirements both forwards and backwards. It is important to ensure it is clear where new requirements came from, but also to be clear where these new requirements are used in the other documents, where appropriate.

Our overall focus for change management was to ensure we used the best of what we had to use. If what we were given was good we kept it, otherwise we changed or replaced it. We wanted to make the very best game that we could that would fit the requirements, and we were not afraid of

making big changes in order to achieve this goal. In our documents new changes were shown using a blue font to make it obvious.

Report updates

GUI report

This document was modified with the following changes:

- Added descriptions for new GUI elements implemented in the game or inherited from Team Farce. We looked at which elements of the GUI we had brought over from our original GUI and also look at the new elements that have been added, such as the journal.
- The report now describes new screens which were added this assessment, the clue found screen and the narrator screen, these are important elements of the finished game so needed to be included in the GUI report.
- Updated descriptions of GUI elements that we brought over from our game
- Expanded content about how user interaction works with the GUI. We felt that our current GUI report did not effectively communicate this, so it was something that we tried to incorporate within the new GUI report. User interaction is an important factor in GUI design that we should not have missed first time round.
- Added a description of our thought process and the factors we considered when designing the GUI

URL:

- <http://docs3.lihq.me/en/latest/Assessment3/gui.html> or
- <http://lihq.me/Downloads/Assessment3/GUI3.pdf>

Testing report

We continued to use the testing document our team created for Assessment 2. This document was modified with the following changes:

- Update references from Assessment 2 to 3
- Update test statistics (number of tests, success rate, number of failures)
- Updated links for Assessment 3
- Added paragraph (2nd from top) explaining how tests work in our project (originally from Team Farce), and how we've adapted and improved their testing methodologies to better fit our workflow. This includes setting up CircleCI for continuous integration. This paragraph also mentions that we kept our own acceptance tests as we also kept our requirements, as well as mentioning that we used and extended upon the unit tests provided with the project we took on.
- Unit test report (see Appendix E) has been updated to reflect the new project and any changes made to the tests. The test results have been added. All unit tests (both added and inherited from Team Farce) have been linked to our requirements, for traceability.
- Acceptance test report (see Appendix D) has been updated to reflect the new project choice, and the results have been updated to reflect upon the latest version of the game. We felt that the acceptance tests weren't specific enough so we updated them to make them more specific, this ensures that the tests can be perfectly replicated. We also tried to remove ambiguous terms such as the word "interact" when referring to picking up clues or speaking to suspects.
- Added reference sources for testing summary, these are placed in the new bibliography
- Added section about automated test coverage, as we started using JaCoCo

URL:

- <http://docs3.lihq.me/en/latest/Assessment3/testing.html> or
- <http://lihq.me/Downloads/Assessment3/Test3.pdf>

Methods and plans

This document was modified with the following changes:

- Added a description of a discussion we had about team management tools. We considered switching to Asana which is a tool similar to GitHub projects to show tasks that need to be done and have discussions, however we felt that it was best to carry on with GitHub projects as it is where our code is and that should ensure we all look at it and keep it up to date.
- Added a description of a debate held about in-person meeting frequency and necessity.
- Described the approach we intend to have when selecting a project for Assessment 4, we will hopefully select one of the three teams that chose our game in Assessment 3
- Added our current thoughts on what we will need to do when we are given new requirements and how we will tackle them, regardless of what they are
- Describes potential task distribution and continuing methods
- Added a detailed description of our methods used in this assessment, including dealing with the new code and the assignment of tasks. We felt it was important that our methods document described the methods used in this assessment, as it gives a good insight into how we worked as a team to tackle this assessment.
- Added a Gantt chart for assessment four to the document to show how we will split up our time for the next assessment
- Updated the appendix which includes the task assignment summary. Each team member was given tasks for Assessment 3

URL:

- <http://docs3.lihq.me/en/latest/Assessment3/methods.html> or
- <http://lihq.me/Downloads/Assessment3/Plan3.pdf>

URL for updated plan: <http://lihq.me/Downloads/Assessment3/AppendixB.pdf>

Risk assessment and mitigation

The risk management document did not change very much, as we were happy with the report from last assessment. Previously our approach and presentation was heavily reworked as described in the previous assessment to include risk ownership.

We looked at the risk management document given to us by the other team but did not really find anything we wanted to add to our own document.

As a team we were very happy with the risk management document that we had and it's format so we decided to keep it. However, we modified this document with the following changes:

- Added database risks to the risk table. When we took on this new project we inherited a new kind of software that we had not previously been using. The other team heavily used a large database, and the use of this new software presented new risks that needed to be looked at. Therefore we discussed these risks as a team and added them into our own document to make the risks specification complete with our new game.
 - One database risk is to do with incorrect data being saved into the database, this could lead to unexpected scenarios when running the game.
 - Another risk is that data is not loaded into our game at all, which could cause the game to not run and would be a major problem with how the game is implemented in this assessment.
- Details and mitigation for both of these risks can be found in the updated document.

URL:

- <http://docs3.lihq.me/en/latest/Assessment3/riskAssessmentAndMitigation.html> or
- <http://lihq.me/Downloads/Assessment3/Risk3.pdf>