

Evaluation and Testing report

Evaluation

The purpose of evaluation is to test whether we have 'made the right product' [1], meeting the needs of the client. While we could do planned testing to determine whether functional requirements had been met, we had to find another method to test our non-functional requirements [2]. This was more difficult, as they are subjective. We also had to ensure that our documentation was up to a high standard.

To find out whether our non-functional requirements had been met, we decided to make a questionnaire [3]. We wrote the questionnaire during a meeting in assessment 4. To make this, we researched software evaluation criteria [4] as well as using our requirements documentation [2] to decide what we needed to assess.

Unfortunately we only got 6 participants to complete the questionnaire which less than the 30 we would need for a sufficient population to form a distribution with. As such we have been unable to make measurements and draw conclusions from the results of the questionnaire. This was due to the extremely limited time frame we had and the prioritisation of completing the rest of the work. Also the demographic of the participants was mainly 19-23 year old university students as this was the main group of people we had easy access to, since the brief asks for a game suitable for use at open days, this is a reasonable demographic as it is likely that mainly 17-19 year old students would play the game at such events. Our results can be found here <http://wedunnit.me/webfiles/ass4/Responses.pdf>.

To assess the quality of the documentation, we scheduled review sessions in the project plan for assessment 4 [5]. During these review sessions we had a checklist which contained a list of requirements we felt were necessary for the document to be complete. The checklist was made near the start of assessment 4, and if a document failed to pass the review session it would be worked on again and have another review session soon after. The checklist containing all of these checklists is available here <http://wedunnit.me/webfiles/ass4/Checklist.pdf>.

Covered in this checklist are the requirements that couldn't be tested in the testing section. We compared our final product with this checklist each time we were confident that it was complete and to standard in a review session. When making the checklist, we searched online for examples and software evaluation criteria [6] and used them as guidance for what we included in ours.

Testing

We performed manual [7] and unit [8] tests to test the final product, as well as testing the code while it was being developed. This practice was continued from the previous group, Team Watson [9]. We also used the same testing methods and practices in the previous sections of the project, as we already had lots of experience with these and knew we could do them well. We also researched different testing methods online to ensure we carried out the appropriate tests [16].

The manual tests were done as black box testing [10]. As part of the testing team was separate from the implementation team, this made it more efficient for us to assume the tester did not have knowledge of the internal structure of the items being tested. Manual testing is good in this regard as it tests to see if the requirements documentation for the software has been met, because a team member acts as an end user. However, manual

testing can be unreliable when done repeatedly. Therefore, doing it each time we made a change would increase the chance of human error. It is also time consuming for the person doing the tests. For this reason, we did manual tests infrequently and mainly when we were choosing which game to take on at the beginning of assessments 3 and 4, and at the end of each assessment in which we were required to code. However, a benefit of manual testing is that we could test the UI of the game, which isn't possible sometimes with automated testing methods.

We made a series of tests (in addition to Team Watson's existing tests) which can be found here <http://wedunnit.me/webfiles/ass4/TestTable.pdf> before performing the manual tests. The table describes what needs to be done by the user for each test and what is considered a passing result for the test. Results are also included in this document, and each test is numbered for referencing purposes. Some of the tests done by Team Watson were now either irrelevant or impossible to perform after the changes we made in assessment 4, so we removed them. These changes can be seen by the removed tests having a line through the text and the text being red and new/changed tests being green text. We also felt that Team Watson hadn't clearly stated what a 'passing' result was for their tests so we have added a 'result after test' column to the manual test table in which we have described what should be expected for a passing result.

The other tests performed by our team members were the unit tests which were done as white box testing [11]. To even write these tests, knowledge of the internal structure of the code is needed. Unit testing is automated. Because of this, unlike manual testing, it doesn't have the issue of being unreliable. However, compared to manual testing, the initial costs of unit testing is higher as they require more time to make. However, after the unit tests are made, they can be rerun so they become cheaper to perform and can be done whenever changes are made to ensure sections of code that have already been written still work. This future-proofs the code, because by seeing which tests fail, we can identify where issues have occurred.

We used the unit tests made by the previous groups and made some new ones for the features we added to the game during assessment 4. They had presented their results as an image [12] showing which had passed and failed. However, we decided this was insufficient as it did not show enough information. Instead, we made a table of unit tests which can be found here <http://wedunnit.me/webfiles/ass4/TestTable.pdf>. This included test names, whether they had passed/failed, descriptions and any other comments we felt were necessary to include. For both manual and unit tests, the previous groups had not shown traceability of the requirements to the testing, which made it more confusing to understand. For this reason, we added a column in each table showing which requirements were related to each test (if any). This kept our formatting consistent with previous assessments [13].

Team Watson also performed play testing [14] during which a user from outside the team plays the game with the purpose of finding bugs. The results from this are in their test table [15]. We decided that we liked this idea but were unsure whether we would have time to do it. Instead, we included a section in our questionnaire from the evaluation section asking the user if they had encountered any bugs and to describe them. This isn't as effective as doing a session of play testing with a user and team member present, but it still allowed us to get feedback, and it didn't require our team members being present.

We adapted Team Watson's requirements [17] by combining them with our requirements [18] from assessment 3. We did this as we prefer our system of laying out the

requirements because we have a column relating user requirements to the relevant system requirements for traceability. This allows us to have a well defined numbering system for all of the requirements, which is extremely useful for referring to them in our documentation.

Have the Requirements Been Met?

The requirements for Assessment 4 can be viewed in our requirements table here <http://wedunnit.me/webfiles/ass4/ReqTable.pdf>. We found that it was easy to determine whether our functional requirements had been met through testing using our test table <http://wedunnit.me/webfiles/ass4/TestTable.pdf>. All of our requirements that were already in the table from previous assessments have been successfully implemented.

However, we decided to remove user requirement 17, which was added into the table by the previous group, Team Watson. This requirement stated that it would only be possible to accuse each NPC once, and made it possible to lose the game if the wrong suspect was accused. However, because the game was scored on a points-based system, we decided this was an unnecessary feature. Accusing the wrong suspect now means the user will lose points, resulting in a lower final score. However, they will still be able to keep playing the game to completion. We decided this would make the game more fun, as part of non-functional requirement 25.

We also added new requirements as part of the Scenario Brief (user requirements 18, 19 and 20), and these were all implemented successfully.

For our non-functional requirements, specifically requirements 25 and 27, we found that it was difficult for us to determine whether or not they had been met, as they are much more abstract. To measure this, we made a questionnaire [3] which we handed out to our target audience (fellow university students) after letting them play our game. However, because we had limited time, we only managed to get 6 responses [19]. This was not enough to have a complete distribution from which to draw a conclusion, which left us unsure about whether or not our non-functional requirements had been met.

We have not managed to meet user requirement 11 as the game we inherited included a lose game system when an incorrect accusation had been made which directly overlaps with said requirement. After removing this feature we couldn't implement the new feature due to time constraints.

Bibliography

- [1] R. Paige, "Validation and Inspection", University of York, 2017.
- [2] H. Cadogan, S. Davison, T. Fox, W. Hodkinson, C. Hughes and A. Percy, "ReqTable.pdf", *Wedunnit!*, 2017. [Online]. Available: <http://wedunnit.me/webfiles/ass4/ReqTable.pdf> . [Accessed: 10- Mar- 2017].
- [3] H. Cadogan, S. Davison, T. Fox, W. Hodkinson, C. Hughes and A. Percy, "Questionnaire", *Wedunnit!*, 2017. [Online]. Available: <https://goo.gl/wXXh4d>. [Accessed: 02- May- 2017].
- [4] K. Stoilova and T. Stoilov, "Software Evaluation Approach", *Virtual ISP*, 2017. [Online]. Available: http://www.visp-project.org/docs/publications/Varna06_4.pdf [Accessed: 06- Mar- 2017].
- [5] H. Cadogan, S. Davison, T. Fox, W. Hodkinson, C. Hughes and A. Percy, "Ass4Plan.pdf", *Wedunnit!*, 2017. [Online]. Available: <http://wedunnit.me/webfiles/ass4/Ass4Plan.pdf> . [Accessed: 12- Mar- 2017].
- [6] Jackson, Mike, Steve Crouch, and Rob Baxter. "Software Evaluation: Criteria-Based Assessment". *Software.ac.uk*. [Online] Available: https://www.software.ac.uk/sites/default/files/SSI-SoftwareEvaluationCriteria.pdf?_ga=1.243965122.671559852.1490182213 [Accessed: 10- Mar- 2017].
- [7] "What is Manual Testing | Goal of Manual Testing | Manual Testing types", *Software Testing Class*, 2017. [Online]. Available: <http://www.softwaretestingclass.com/what-is-manual-testing/> [Accessed: 05- Mar- 2017].
- [8] "Unit Testing – Software Testing Fundamentals", *Softwaretestingfundamentals.com*, 2017. [Online]. Available: <http://softwaretestingfundamentals.com/unit-testing/> [Accessed: 05- Mar- 2017].
- [9] Beldie. C. et al. "Team Watson Assessment 3 Change Report" [Online] Available: <http://teamwatson.webs.com/Change3-1.pdf> [Accessed: 05- Mar- 2017].
- [10] "Software Testing - Quick Guide", *www.tutorialspoint.com*, 2017. [Online]. Available: https://www.tutorialspoint.com/software_testing/software_testing_quick_guide.htm [Accessed: 11- Mar- 2017].
- [11] "Black Box Testing – Software Testing Fundamentals", *Softwaretestingfundamentals.com*, 2017. [Online]. Available: <http://softwaretestingfundamentals.com/black-box-testing/> . [Accessed: 08- Mar- 2017].
- [12] "White Box Testing – Software Testing Fundamentals", *Softwaretestingfundamentals.com*, 2017. [Online]. Available: <http://softwaretestingfundamentals.com/white-box-testing/> [Accessed: 17- Jan- 2017].
- [13] "Team Watson Unit Test Results", *Teamwatson.webs.com*, 2017. [Online]. Available: <http://teamwatson.webs.com/Testing%20stuff.png>. [Accessed: 07- Mar- 2017].
- [14] H. Cadogan, S. Davison, T. Fox, W. Hodkinson, C. Hughes and A. Percy, "Tests.pdf", *Wedunnit!*, 2016. [Online]. Available: <http://wedunnit.me/webfiles/ass2/Test-table.pdf> . [Accessed: 10- Mar- 2017].
- [15] "What is Playtesting? - Definition from Techopedia", *Techopedia.com*, 2017. [Online]. Available: <https://www.techopedia.com/definition/27197/playtesting> [Accessed: 05- Mar- 2017].

- [16] Beldie. C. et al. "Team Watson Assessment 3 Test Table" [Online] Available: <http://teamwatson.webs.com/testing%20table.pdf> [Accessed: 05- Mar- 2017].
- [17] Beldie. C. et al. "Team Watson Assessment 3 Requirements" [Online] Available: <http://teamwatson.webs.com/Updated%20requirements-1.pdf> [Accessed: 03- Mar- 2017].
- [18] H. Cadogan, S. Davison, T. Fox, W. Hodgkinson, C. Hughes and A. Percy, "Requirements Table Assessment 3", *Wedunnit!*, 2017. [Online]. Available: http://wedunnit.me/webfiles/ass3/Requirements_Table_Ass3.pdf [Accessed: 03- Mar- 2017].
- [19] H. Cadogan, S. Davison, T. Fox, W. Hodgkinson, C. Hughes and A. Percy, "Questionnaire Responses", *Wedunnit!*, 2017. [Online]. Available: <http://wedunnit.me/webfiles/ass4/Responses.pdf> [Accessed: 02- May- 2017].