Software Review Plan

Rappid app development

LNL

Table of Contents

[How the team ensured the development quality of the software during the three sprints (test plans, etc). 2](#_Toc44058563)

[What processes were utilised to ensure adequate control of the quality of the development process. 2](#_Toc44058564)

[Describe the code testing process. 2](#_Toc44058565)

[How can the software be supported for future modification and refinements (comments, code documentation, development plans)? 2](#_Toc44058566)

[Provide a mapping of the user requirement to the final software product (did you meet the client’s expectations and requirements). 2](#_Toc44058567)

# 1 How the team ensured the development quality of the software during the three sprints (test plans, etc).

The team ensured coding standards were followed during all sprints allowing for code readability through proper formatting and commenting.

Test plans were performed at each sprint to ensure that the key requirements of each sprint were met and working.

Thorough meetings with the client allowed us to see what he wanted and what was required for us to.

# 2 What processes were utilised to ensure adequate control of the quality of the development process.

Uploading different test versions to GitHub for further review to see if they’re functioning and compatible with what code base we have already established for each previous week.

Reviewing each week’s previous tasks and adding on to them. Each week we have done a test case and followed a Gantt with a to do list on trello.

# 3 Describe the code testing process.

The code testing process used by LNL goes as follows.

The tester establishes a test case; gives it a description, expected outcome and Evidence to prove that all is working with the item.

Below is an example of one test case from sprint 3

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Description | Expected Outcome | Evidence |
| Case 1 | Creating an admin using an incorrect password | Error given |  |

As it can be seen above, we keep it simple by testing key areas of the site. We write the code, ensure functionality through a lack of syntax and logic errors, then finally record that it is working in a test case.

# 4 How can the software be supported for future modification and refinements (comments, code documentation, development plans)?

1. Writing code with up to date standards
2. Documentation of development plans
3. Comments on code
4. Version control
5. Read me / user documentation

# 5 Provide a mapping of the user requirement to the final software product (did you meet the client’s expectations and requirements).

## Business Requirement Identification

|  |  |
| --- | --- |
| Code | Requirement Description |
| BR1 | The website must be responsive or adaptive |
| BR2 | Users sign up for a weekly or monthly mailing list |
| BR3 | Users can request to leave the mailing list |
| BR4 | Create a membership page to display user info |
| BR5 | Admin can remove user from mailing list |
| BR6 | Top 10 history calculated when a user adds a rating to a movie |
| BR7 | Show Top 10 history |
| BR8 | Password requirements for admin account |
| BR9 | Create an admin |

## Requirements Traceability Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| Business Requirement | Test Case # | Defects | Status |
| BR1 | 10, 11, 12 (RAD Sprint 1) | - | Satisfied |
| BR2 | 2, 5, 7, 8 (RAD Sprint 2) | - | Satisfied |
| BR3 | 9 (RAD Sprint 2) | - | Satisfied |
| BR4 | 11, 12 (RAD Sprint 2) | - | Satisfied |
| BR5 | 13 (RAD Sprint 2) | - | Satisfied |
| BR6 | 3 (RAD Sprint 3) | - | Satisfied |
| BR7 | 4 (RAD Sprint 3) | - | Satisfied |
| BR8 | 1 (RAD Sprint 3) | - | Satisfied |
| BR9 | 2 (RAD Sprint 3) | - | Satisfied |

So far, every user requirement as been met at the meetings with the client. On occasion he has requested some tweaks or extra requirements, we have been more than happy to meet these requirements of his as long as they were within reason and would not cause scope creep.