KASPER – ID5 PROCESS DOCUMENTATION

SOFTWARE DEVELOPMENT TEAM:

Project Manager: Tushita Patel

Dev Lead: Kristof Mercier, Dylan Prefontaine

Test Lead: Jeremy Liau

Build Manager: Christopher Mykota-Reid (ChrisMR)

Developers: Gaurav Arora, Haotian (Justin) Ma, Melody (Tian) Zhao

Test Team: Christopher May (Chris May), Ryan Tetland

Documentation: Arianne Butler

Contents

1.0 Task Assignments

Dev Team

- Below are the links to our front and back-end Trello boards, respectively. These outline the ID5 task assignments for the dev team. You will be prompted to sign in to view these boards. Each card is accompanied by a time estimate as approximated by the dev leads.

  Front-end: https://trello.com/b/S2XiqLAm/dev-id4-front-end

  Back-end: https://trello.com/b/m9OtfvS1/dev-id4-back-end

Test Team

- Bug Party 2: Everyone
- Hallway and Acceptance testing: Jeremy
- Defect estimation: Tushita, Jeremy
- Bug fixing with developers: Jeremy, Ryan, Chris May
- Code policing (Goal: 75% coverage): Ryan, Chris May
- Front end smoke test fixing: Ryan, Chris May

Documentation

- Organization, compilation, and editing of documentation: Arianne

2.0 Activity Log

The following link leads to our term schedule. Click on the "**Individual Activity Log**" tab at the bottom for our group activity log, as well as the group member contribution pie chart. This section also indicates which tasks were completed via pair programming. Click on the "**Activity Log Totals**" tab to see the summary of hours for each group member. In the "Activity Log Totals" sheet, you will also find a section that displays the percentage of hours worked that have been peer reviewed by at least one other group member. Currently about 75% of hours worked has been peer reviewed.

Note: we are currently experiencing difficulty with this link and you may need to copy and paste it into your web browser.

https://docs.google.com/spreadsheets/d/1g3CfvlI1erdg2zaHZlVyh6uQ-rYQ6L5Cay0YIn2UETk/edit#gid=0

3.0 Bug Party

Date: Wednesday, March 15th /17 @3:30 p.m.

Location: S371 and S372

The second bug party was held at the start of ID5, right after the feature freeze. This allowed us to dedicate ID5 to bug fixes and clean-up. The team split in two with a fair distribution of testers and developers.

A Bug Party Expectation Document was provided to all team members priori to the bug party. Two Trello Boards (linked below) were created for the two groups to report and track their bugs.

https://trello.com/b/qoRs0fZe/bug-party-ii-s371

https://trello.com/b/hr5Kx51d/bug-party-ii-s372

Discovered STI Format:

The format of the report (to be placed in the Trello card) is:

Title:
Priority: (rate 1-4, 4 being high)
Severity: (Data loss, Major functionality, or Minor functionality)
Reproduction formula: (Step by step how to get there)
Area of the project: (Front-end, back-end, both, or unknown)
Person opened by: (Your name)

Trello Board Bug Categories:

1) Important - All sanitized bugs will be moved to this list after the bug party by the test lead.
2) In progress - All sanitized bugs deemed high priority and high severity.
3) Resolved - Fixed bugs are moved here by the developer once they think that they have fixed the problem.
4) Closed - The testers should move the fixed bugs to the Closed stage once they agree with the resolution and decide to close it.

Bug Party Statistics:

- Group 1 discovered 25 STIs, of which 21 were sanitized.
- Group 2 discovered 32 STIs, of which 25 were sanitized.
- There was a total of 17 overlapping STIs, of which 13 were overlapping sanitized bugs.

Bug Estimate:

Bug Estimation Formula: (Number of Defects from Group A) * (Number of Defects from GroupB) / number of overlaps

= 21 * 25 / 13
~ 40 defects

4.0 Risk Assessment

Introduction

Risks are divided into two categories – technical and non-technical. Technical risks are related to the construction and design of our code, and non-technical risks relate to team management, such as group structure and client communication. For each risk identified, this risk assessment will provide estimates for probability of occurrence and severity, possible scenarios that could cause the risk to materialize, and mitigation and contingency plans. This report also contains a section dedicated to materialized risks and their effect on our project.

4.1 Technical Risks

4.1.1 Not All Bugs Are Fixed by the End of Term

Probability: 0.8
Loss: 0.3

Scenario: There are too many bugs in our code to fix before the end of term. There may also be undiagnosed bugs in the system that cannot be fixed, despite best efforts.

Mitigation: A bug party will be held early in ID5 to find as many bugs as possible and to estimate the number of bugs that exist in the code. This estimate will include known and unknown bugs, and will allow the team to allot time for as many bug fixes as possible.

Contingency Plan: There is a high probability that not all bugs can be fixed in time. All known bugs that are not fixed will be well documented for future developers in Programmer Documentation.

4.1.2 Bug Fixes Introduce New Defects

Probability: 0.3
Loss: 0.3

Mitigation: Smoke tests will be designed to check for dependencies between the front and back-end. This will decrease the likelihood of introducing new problems. Regression tests will be run ensure that changes do not reintroduce any old bugs.

Contingency Plan: If a bug fix introduces any new defects, the team will decide if the fix is high priority. If the fix is low priority, it will be documented in the defect reports in Programmer Documentation and ID5 Project Documentation.

4.1.3 Found bugs are Not Reproducible

Probability: 0.2
Loss: 0.4

Mitigation: Testers will plan their tests ahead of time and document exact steps. This will verify whether the bug is reproducible via a fixed set of steps.

Contingency Plan: If a bug is non-reproducible and cannot be fixed, it will be documented.

4.1.4 Not Enough Code is Peer Reviewed

Probability: 0.6
Loss: 0.2

Scenario: Group members do not hold pair programming sessions enough resulting in less reviewed code. This could lead to a decrease in quality and/or additional bugs.

Mitigation: Group members are required to complete at least one pair programing session per ID, and will be encouraged to do more. All group members are also required to hold one code review during the term. An excel spreadsheet is set up to record and monitor what percentage of completed work has been peer reviewed.

Contingency Plan: If the percentage of peer reviewed work is too low, more people will be assigned to Code Police. Our goal is to have 75% of our code reviewed by end of term.

4.2 Non-Technical Risks

4.2.1 Client Unavailability Results in Misunderstanding of Requirements

Probability: 0.3
Loss: 0.5

Scenario: Our client is away for most of ID5 and client meetings will not take place. All requirements decisions will be made by the team and could therefore conflict with the client's wishes.

Mitigation: Communication through email may be possible. If not, we will continue work based on the client's previous requirements.

Contingency Plan: There is no contingency plan for this risk, because there is no further opportunity to handle misunderstandings before the end of term.

4.2.2 Misunderstanding with Client over Code Ownership

Probability: 0.3
Loss: 0.1

Scenario: Our client expects to acquire and own the source code because it was his idea.

Mitigation: This risk will be discussed with Dr. Osgood to resolve ownership issues. A discussion with our client will take place to ensure he is aware of these issues.

Contingency Plan: If issues with our client cannot be resolved, a meeting will be set up with Conrad, Dr. Osgood, and various team members to work out a solution.

4.2.3 Future Developers have Problems with Our Code

Probability: 0.6
Loss: 0.2

Scenario: If the project is continued by future developers, they may not understand certain tools and frameworks used in our project.

Mitigation: Programmer Documentation will be created to explain all development, build, and testing tools used for our project. The developers and testers have been encouraged to clearly comment their code.

Contingency Plan: Programmer Documentation will cover as many areas as possible, including installation instructions, tricky areas of code, and explanations regarding integral pieces of the system.

4.2.4 Busy Schedules

Probability: 0.8
Loss: 0.3

Scenario: In ID5, many group members will be busy with other projects. This will result in less productivity.

Mitigation: All group members are expected to complete their individual tasks. Communication regarding availability will result in more realistic goals. The main priority for ID5 is documentation.

Contingency Plan: If group members cannot finish expected tasks, they will be cut from the project and documented.

4.3 Materialized Risks

The following section describes the ID5 materialized risks. Each materialized risk will be accompanied by a description, a plan for resolution, and its effects on the project.

Busy Schedules

Description: Productivity of our group was down in ID5. This is a result of end-of-term.

Resolution: Documentation was prioritized because it is important for future development of the project.

Effects: Some functionality was cut from the project and documented.

5.0 Meeting Notes

There were 2 group meetings for ID5, and frequent stand-ups both before class and via Slack. The following link contains our documented meeting notes and pre-class stand-ups:

https://github.com/CMPT371Team1/Documentation/blob/master/Meetings/371-MeetingNotes.docx

For frequent online stand-ups, see our “stand-ups” channel on Slack:

https://cmpt371group1.slack.com/messages/C43K3962J/