***Gamification of Data***

***S/W Quality Assurance***

**Senior Design Project - CIS 4951 - Fall 2019**

**Version No. 1.0**

Team Members:

Ryan Ross (Team Lead)

Cameron Kozan

Zaid Alsafi

Alex Navarre

**Project Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version Number | Date | Revising Author | Description of Revision |
| 1.0 | 11/15/2019 | Cameron Kozan | Document Created |

|  |
| --- |
| **Table of Contents** |

[**1.0 Introduction**](#_83ik91ktkl1p) **5**

[1.1 Scope and intent of SQA activities](#_qj8qi8q97e9j)…………………………………………………………………………. 5

[1.2 SQA organizational role](#_nx96034zbpcp)…………………………………………………………………………………….. 5

[**2.0 SQA Tasks**](#_ga2s9lv0alkz) **5**

[2.1 Task Overview](#_xs8lmbmpwck6)………………………………………………………………………………………………. 5

[2.1.1 Description of SQA Task 1](#_7gakxcpzsvkh)…………………………………………………………………………... 5

[2.1.1.1 Work Products and Documentation for Task 1](#_2wcamxdi2sch)………………………………………………… 5

[2.1.2 Description of SQA Task 2](#_g1eddvlnlbow)…………………………………………………………………………... 6

[2.1.2.1 Work Products and Documentation for Task 2](#_ejwrfi109rxq)……………………………………………….... 6

[2.1.3 Description of SQA Task 3](#_mnr9jxt7fkxp)…………………………………………………………………………... 6

[2.1.3.1 Work Products and Documentation for Task 3](#_c1cbi9rod3br)………………………………………………… 6

[2.2 Standards, Practices and Conventions (SPC)](#_9qvnevxeofig)................................................................................................... 6

[2.3 SQA Resources](#_s563n8cpzmew)……………………………………………………………………………………………... 7

[**3.0 Reviews and Audits**](#_doj7355pi2av) **7**

[3.1 Generic Review Guidelines](#_s0aixd8g1cv5)…………………………………………………………………………………. 7

[3.1.1 Conductions a Review](#_frino6xoc397)………………………………………………………………………………...[7](#_frino6xoc397)

[3.1.2 Roles and Responsibilities](#_langnz1a4h5k)……………………………………………………………………………. 7

[3.1.3 Review work products](#_6ra529il3yc0)………………………………………………………………………………... 8

[3.2 Formal Technical Reviews](#_ng0dgmzbgy0d)…………………………………………………………………………………... 8

[3.2.1 Description of Software Project Management Plan review](#_9ahxudvyzt34)…………………………………………... 8

[3.2.1.1 Description and Focus of the review](#_qaifc022o8c1)…………………………………………………………... 8

[3.2.1.2 Timing of the review](#_ftvmk9tvh5vm)…………………………………………………………………………... 8

[3.2.1.3 Work products produced](#_146qxin8rud5)……………………………………………………………………….. 8

[3.2.1.4 Review checklist](#_2eb6sevr9h0h)……………………………………………………………………………….. 8

[3.2.2 Description of RMMM review](#_g18ilaapq9md)……………………………………………………………………….. 9

[3.2.2.1 Description and Focus of the review](#_c0m00qsvc0gm)…………………………………………………………... 9

[3.2.2.2 Timing of the review](#_duzx937xhhcl)…………………………………………………………………………... 9

[3.2.2.3 Work products produced](#_v53n0osiag0b)……………………………………………………………………….. 9

[3.2.2.4 Review checklist](#_f142b7haqhc6)……………………………………………………………………………….. 9

[3.2.3 Description of Requirements review](#_fhsbch5a9ubn)………………………………………………………………... 10

[3.2.3.1 Description and Focus of the review](#_ll5grk9koaa)…………………………………………………………. 10

[3.2.3.2 Timing of the review](#_vghngcsiznth)…………………………………………………………………………. 10

[3.2.3.3 Work products produced](#_28pya52y069f)…………………………………………………………………….... 10

[3.2.3.4 Review checklist](#_hg2362oxyyeo)…………………………………………………………………………….... 10

[3.2.4 Description of Architectural design review](#_6jbcdijaux63)…………………………………………………………. 11

[3.2.4.1 Description and Focus of the review](#_e1hgdvkbbp63)......................................................................................... 11

[3.2.4.2 Timing of the review](#_qobc91u7wnha)………………………………………………………………………….. 11

[3.2.4.3 Work products produced](#_c7e5ks1qukt1)…………………………………………………………………….... 11

[3.2.4.4 Review checklist](#_fcnjyfnfu9p7)…………………………………………………………………………….... 12

[3.2.5 Description of Interface design review](#_cug2wm3vabft)…………………………………………………………….... 12

[3.2.5.1 Description and Focus of the review](#_nzigpleb1fgh)…………………………………………………………. 12

[3.2.5.2 Timing of the review](#_wzauokhbwbvf)…………………………………………………………………………. 12

[3.2.5.3 Work products produced](#_incwcp7w2wc)…………………………………………………………………….... 12

[3.2.5.4 Review checklist](#_gecxs35twxsq)…………………………………………………………………………….... 12

[3.2.6 Description of Code review](#_w1o4zei0bpej)…………………………………………………………………………. 13

[3.2.6.1 Description and Focus of the review](#_g0ins6f1cijh)…………………………………………………………. 13

[3.2.6.2 Timing of the review](#_kanfcv47021d)…………………………………………………………………………. 13

[3.2.6.3 Work products produced](#_ereixkt4kyy0)…………………………………………………………………….... 13

[3.2.6.4 Review checklist](#_m2b07qn9teqd)……………………………………………………………………………... 13

[3.2.7 Description of Test specification review](#_p4wqgz28ui1e)……………………………………………………………. 13

[3.2.7.1 Description and Focus of the review](#_p5i6ea71b6su)…………………………………………………………. 13

[3.2.7.2 Timing of the review](#_uc5orz2vg0uf)…………………………………………………………………………. 14

[3.2.7.3 Work products produced](#_oy6qof19des5)……………………………………………………………………... 14

[3.2.7.4 Review checklist](#_2dq3q1xt1c2o)……………………………………………………………………………... 14

[3.3 SQA audits](#_h6qccd6ppemp)…………………………………………………………………………………………………. 14

[**4.0 Problem Reporting and Corrective Action**](#_3uayrkxx35rw) **14**

[4.1 Reporting Mechanisms](#_jxwrgmeo9kwt)…………………………………………………………………………………….. 14

[4.2 Responsibilities](#_f6yhp5j5ki8v)…………………………………………………………………………………………….. 14

[4.3 Data collection and evaluation](#_nm82ojbc0g2j)……………………………………………………………………………... 15

[4.4 Statistical SQA](#_4lo56he9rmst)……………………………………………………………………………………………... 15

[**5.0 Software Process Improvement Activities**](#_azyxqjtbl0dw) **15**

[5.1 Goal and Objectives of SPI](#_ju1xs7ij6fu6)……………………………………………………………………………….... 15

[5.2 SPI tasks and responsibilities](#_cvgsd46eigug)………………………………………………………………………………. 15

[**6.0 Software Configuration Management Overview**](#_ja4nvhb0guir) **16**

[6.1 Overview](#_u0ou7t9633pc)……………………………………………………………………………………………………. 16

[6.2 Approach](#_xxlqcsm5m5d)……………………………………………………………………………………………………. 16

[6.3 Impact](#_7nifktkifar2)………………………………………………………………………………………………………. 16

[**7.0 SQA Tools, Techniques, Methods**](#_4di8q3ibc3j8) **17**

[**8.0 Appendix**](#_wo1mt58oc1qb) **17**

# **1.0 Introduction**

## **1.1 Scope and intent of SQA activities**

The team’s objective is to ensure that the product does not deviate far from the original gamification design specifications. If it is discovered that deviation has occurred, the team will notify each member to prevent future deviations and to correct the previous deviations. Also, the team will perform a walkthrough to analyze the product’s quality at any stage of development. Error detection and possible enhancements are also expressed to each team member.

## **1.2 SQA organizational role**

The team organizational role is to review the product at specific times during product implementation. Upon reviewing, the team member duties will be to evaluate the software at its current development stage and recognize any defects in the subsequent stage (design or implementation). The team will directly interact with the customer in group discussions, discussing any errors or possible enhancements that have been identified. In addition, the team will ensure that the customer has not deviated in any way from the initial design specifications.

# 2.0 SQA Tasks

## 2.1 Task Overview

### 2.1.1 Description of SQA Task 1

The team will meet with Sargon Partners every other week to adjust requirements if need be. Every week, the lead software engineer will match the software to the current design requirements agreed upon at meetings. This process will help ensure that the software development will be handled correctly. The weekly checks will consist of matching the requirements to individual segments of code and assessing their correctness.

#### 2.1.1.1 Work Products and Documentation for Task 1

As a result of Task 1, internal meeting minutes will be taken during meetings and reviewed at internal meetings. The meeting minutes serve as a solid baseline for our team meetings.

### 2.1.2 Description of SQA Task 2

The design lead will develop initial mockups for the 8\* individual pages that the website is required to have. Two pages of mockups must be finished for each meeting with our clients to assess correctness and make potential changes to better fit the overall requirements. The lead will distribute the work evenly between group members and hold individual meetings to address small issues and keep the design unified.

#### 2.1.2.1 Work Products and Documentation for Task 2

As a result of Task 2, the rough drafts/mockups will be saved in our design requirements appendix.

### 2.1.3 Description of SQA Task 3

Each group member will have to periodically go through the website to perform reliability testing. Individual use should help our team find the smaller, potentially lower priority, errors early on. Performing periodic testing throughout the life of our project will help ensure we can mitigate problems closer to when they arise. Each member will be responsible for checking new functionality, along with existing functions.

#### 2.1.3.1 Work Products and Documentation for Task 3

As a result of Task 3, a short report will be written for our client meetings. The reports will cover issues found, their priority, and resolution. The reports will be collected in a defect log.

## 2.2 Standards, Practices and Conventions (SPC)

Documentation Standards: Whenever applicable, documents will all follow the same format and layout. This layout was predetermined by our group before the project began.

Coding Standards: Standard AWS RDS naming conventions and format will be followed for the database. HTML5/CSS3 standards will be upheld for the website creation.

We will also be following both C# and Java best practices.

## 2.3 SQA Resources

* Eclipse IDE – Java testing and development
* Visual Studio Code – HTML testing and development
* Visual Studio Community – C# testing and development
* AWS RDS – database development and testing
* Microsoft Azure – database development and testing
* Figma – page design mockups
* Google Drive – collaborative document management
* Test.AI - Automated Web testing
* Selenium - Automated Web testing
* Cypress.io - Automated Web testing

# **3.0 Reviews and Audits**

## 3.1 Generic Review Guidelines

### 3.1.1 Conductions a Review

The software will have scheduled reviews to detect any defects in the current prototype, and to determine any notable enhancements that should be implemented prior to the final product. The current prototype will be broken down into smaller subsets and each subset will be reviewed. The team will attend each review and critique each other’s part of the software to ensure that the maximum number of possible defects are accounted for. The group will meet on a weekly basis to discuss defects or enhancements. Each meeting will consist of no more than a couple of hours. The current prototype for both the user-interface and the back-end will be available at every meeting allowing defects to be explicitly pointed out to each group member. If the defect found could not be easily duplicated, the Development leader will take note of the defect. Any desired enhancements will be discussed to determine which are necessary and feasible. Any enhancement found to be too difficult or unnecessary for product completion will be noted by the Development leader.

### 3.1.2 Roles and Responsibilities

The Development leader will oversee any formal technical reviews. Any defects or enhancements will be discussed and recorded by the Development leader. Each defect or enhancement will be given a priority rank, which will be recorded. Once the review is complete, the Development leader will make a summary of each defect or enhancement and distribute them to the appropriate member of the team.

Each member will be responsible for reviewing his own software module during module creation and upon module completion. Once each major software module is complete, it is the members duty to inform the Development leader that the module is ready for review.

### 3.1.3 Review work products

The Development leader will keep a defect log. The defect log contains all defects and enhancements, as well as a priority rank for each. The following will also be noted in the defect log: (1) Priority level of the defect, (2) whether the defect or enhancement has been handled, (3) which software engineer oversaw the correction, and (4) what date the correction was completed.

## 3.2 Formal Technical Reviews

### 3.2.1 Description of Software Project Management Plan review

#### 3.2.1.1 Description and Focus of the review

The software project management plan review will focus on insuring the overall scope of the project is correct and that any updates that need to occur will happen and be documented correctly will all parties signing off on any and all scope changes.

#### 3.2.1.2 Timing of the review

The document will be reviewed at the start of the second semester or when the client requests a change in the scope of the project.

#### 3.2.1.3 Work products produced

The SPMP will be updated and correct versioning will be observed by storing a copy of the old document for review purposes. This will be an additional document created to layout all changes that took place and why. This will keep all parties informed on the current scope.

#### 3.2.1.4 Review checklist

* Review old SPMP for possible updates
* Talk with clients about any changes to project scope and document all in separate document to track needed changes
* Version and update new document
* Get signoff by team lead and client

### 3.2.2 Description of RMMM review

#### 3.2.2.1 Description and Focus of the review

The RMMM review will focus on determining if the proposed risk management for the development of this software is within reason. The focus will be on if the risk management can handle a proposed problem accordingly. If the RMMM document is not managing each risk accordingly, the RMMM will be amended to correct the oversight.

#### 3.2.2.2 Timing of the review

The RMMM review will be held upon completion of the RMMM document. This should occur within the first few weeks of the software’s development. If any kind of updates are needed for the RMMM document all team members will inform each other, and when the document is updated it will be reviewed by the team.

#### 3.2.2.3 Work products produced

The team leader will create a summary report of the RMMM review. This includes any possible risks that have not been covered, and any risks that have been accounted for, but are not managed appropriately. Once a new risk is proposed, a discussion will take place on how to handle the risk if it were to occur. Any risks being managed inappropriately will also be discussed and an amendment to their management will be made to better handle the risk.

#### 3.2.2.4 Review checklist

* Have all risks been thoroughly covered in the document? If not, what is missing?
* Of the risks covered in the document, are there any that did not seem to be effectively covered? If yes, which one(s)?
* Of the risks covered in the document, are there any that did not seem to be appropriately managed? If yes, which one(s)?

### 3.2.3 Description of Requirements review

#### 3.2.3.1 Description and Focus of the review

The Requirements Specification review will work to analyze the proposed design of the software. The focus of this review will be to remove or discuss changes to any obvious design flaws. Once a design defect has been encountered, the Development team will discuss with the other team members for any ideas or suggestions about how to compensate for the design flaw.

#### 3.2.3.2 Timing of the review

The Requirements Specification review will be held upon completion of the Requirements Specification. This should occur within the first few weeks of the software’s development. To ensure that the software is conforming to the restrictions of the design, each team members will frequently conduct his own Requirements Specification review. If a team member determines that the current design of a module is flawed, it will be brought to the attention of the Design leader and an appropriate discussion to amend the problem will be held.

#### 3.2.3.3 Work products produced

The Document leader will create a summary report of the Requirements Specification Review. This includes any defects or enhancements that have been brought to attention. Once design defects have been identified, the Document team will discuss possible solutions with the team members. Each possible solution will be noted and reviewed again to determine if the solution will have an impact on the rest of the design. Once all obvious design defects have been handled, the Requirements Specification will be amended to account for the design changes.

#### 3.2.3.4 Review checklist

* Is the proposed design the best possible solution?
* Is there any obvious design flaws that have not been accounted for? If yes, what?
* Are there any necessary enhancements for the software?
* Is the proposed Requirements Specification within the time frame?

### 3.2.4 Description of Architectural design review

#### 3.2.4.1 Description and Focus of the review

The Architectural Design review will provide a basis for analysis of the proposed architectural design of the software. The focus of this review will be to assess the current design to ensure that data flow and data control are being handled appropriately. If a design flaw has been discovered, the team leader will discuss with the team members of any ideas or suggestions about how to compensate for the architectural design flaw.

#### 3.2.4.2 Timing of the review

The Architectural Design review will be held upon completion of the Requirements Specification and the System Specification. This should occur within the first few weeks of the software’s development. This is necessary to ensure that the underlying architectural design of the software is sound and will not create problems for the software engineers or degrade the software’s performance in the future.

Each team member will frequently conduct his own Architectural Design review to ensure that the software is conforming to the restrictions of the architectural design. If the team member determines that the current architectural design of a module is flawed, it will be brought to the attention of the team leader and an appropriate discussion to amend the problem will be held.

#### 3.2.4.3 Work products produced

The team leader will create a summary report of the Architectural Design review. This includes any defects in the architectural design that has been brought to attention. Once architectural design defects have been identified, the team leader will discuss possible solutions with the team. Each possible solution will be noted and again reviewed to determine if the solution will have an impact on the rest of the design. Once all obvious architectural design defects have been handled, the Requirements Specification and System Specification will be amended to account for any architectural design changes.

#### 3.2.4.4 Review checklist

* Is the proposed architectural design the best possible solution?
* Is there any obvious architectural design flaws that have not been accounted for (i.e. slow data flow or control)? If yes, what?
* Are there any obvious changes you see would further enhance the software’s performance?
* Is the proposed architectural design complete? If not, what seems to be missing?
* Does the proposed architectural design seem possible within languages of choice? If not, why?

### 3.2.5 Description of Interface design review

#### 3.2.5.1 Description and Focus of the review

The Interface design review will be conducted first will team members responsible for the design, look and feel for the website. After the initial review, another review will take place with the clients to insure they are happy with the design. This allows for quick design with good feedback before full implementation.

#### 3.2.5.2 Timing of the review

The first part of the review will be completed by the team members before a meeting with the clients. The review is then complete at the client meeting which occur on a weekly basis.

#### 3.2.5.3 Work products produced

The interface mockups created will be living documents until implementation for that specific interface. This will allow for quick changes to the designs.

#### 3.2.5.4 Review checklist

* Does the design include all needed functionality?
* Is it pleasing to look at?
* Is the color scheme consistent across designs?
* Is it possible to implement?

### 3.2.6 Description of Code review

#### 3.2.6.1 Description and Focus of the review

The code review is to mainly spread knowledge about a piece of functionality. In the code review teammates are allowed to ask questions, give insight on how code can be written better and even rewrite code altogether.

#### 3.2.6.2 Timing of the review

The reviews will happen every fortnight at our meetings on Wed. If a member is not able to come to the code review, they will be required to complete a code review on their own time, within a 24 hour period.

#### 3.2.6.3 Work products produced

The work products that will be produced from the code reviews are that we will be able to work on the next thing on the roadmap.

The code review will be the last thing that happens to a piece of code before it gets delivered. Once a piece of code’s code review is done, the person whom created the code will be put on a different piece of functionality.

#### 3.2.6.4 Review checklist

* is the source code reliable ? If not, why?
* Is the source code efficient? If not, why?
* Is the source code highly modular?
* Is the source code easy to read?
* Is the source committed well?
* Are they using any bad practices?
* Is there a simpler way?
* Is the source code allow for easy maintainability?

### 3.2.7 Description of Test specification review

#### 3.2.7.1 Description and Focus of the review

The test specification review will be used to review and analyze our four step test strategy. The focus of this review will be to ensure we follow our individual test strategies. The review period will be conducted before each test stage.

#### 3.2.7.2 Timing of the review

The review period will take place before and after each stage of our test strategy: backend, frontend, integration, and regression testing. The initial review will be used to ensure our test cases are still valid after changes and alterations have been made. The post review will be to go over the test results.

#### 3.2.7.3 Work products produced

The work products produced from the Test specification review will be insight into how to improve our following test specification reviews for the upcoming test strategies.

#### 3.2.7.4 Review checklist

* Are the test specifications being met? If not, why?
* Is the backend being fully tested? If not, why?
* Is the frontend being fully tested? If not, why?
* Is the integration being fully tested? If not why?
* Is the regression testing addressing everything? If not, why?
* Is the test specification review helpful for further testing? If not, why?

## 3.3 SQA audits

We believe that there should be a “head” of SQA to oversee all of the SQA activities. We have designated that to the team lead. The head of the SQA will be in charge of making sure that version numbers are being updated, maintaining order during code reviews and automated tests are being created.

# **4.0 Problem Reporting and Corrective Action**

## 4.1 Reporting Mechanisms

For all minor issues that occur in any FTR, that issue will be reported to every member on the team for feedback on how to fix the issue and proceed with the task at hand. If a large issue occurs the issue will be reported to all team members and the client in order to use all the resources available to fix the issue that arose.

## 4.2 Responsibilities

The responsibility to contact the team will be the member that completes the FTR audit. At that point the team will decide if the issue is large enough to escalate the issue to the client. The team lead will send an email to inform the client of any important information at that point.

## 4.3 Data collection and evaluation

All error/defect data will be collected at the time it occurs whether the team member deems it necessary or not. The data will be stored in the review document if one exists or a document will be created to store the data along with any information about how the error occurred.

## 4.4 Statistical SQA

If the error that occurs is recurring then a basic time series analysis will be performed to gain as much insight about the issue as possible. Bar charts may be used to show trends in the data.

# **5.0 Software Process Improvement Activities**

## 5.1 Goal and Objectives of SPI

The goals and objectives of the SPI are to lower the frequency of software defects and to determine the underlying cause of the defects that occur. Furthermore, once the underlying causes have been identified, measures will be taken to determine the best course of action to eliminate the problem.

## 5.2 SPI tasks and responsibilities

Based on the Statistical information gathered, the Development leader will keep a tally of what errors or causes of errors occur most frequently. The more often a defect occurs from the same underlying cause, the more problematic an area will be considered. Depending on the nature of the cause and the individuals involved, two actions can take place: (1) the Development leader will investigate the Statistical SQA information and the defect log to determine if the problem area exists primarily for a single developer, or (2) if every developer experiences the same problem.

If the problem occurs mostly from one user, the developer will be informed of the frequency of the personal problem area. Most often, the developer has a better idea of how to handle his own implementation oversights.

If the problem occurs from all developer, development practices will be analyzed to determine the cause of the problem. The problem and possible solutions will be examined at a meeting to all the developers.

# **6.0 Software Configuration Management Overview**

## **6.1** **Overview**

We will be creating a baseline for each artifact to maintain version control and to be able to view what changes have been made, along with seeing whom was responsible for each of these changes. There will be a project roadmap which will have all of the dates to which each configuration item shall be delivered. These dates have been agreed upon by both the team and the clients. If the dates wish to be changed,it MUST be agreed upon by both the team and client. Last but not least, we will have code reviews to make sure that the code we write is complete, consistent and is well written. In order for a code review to be deemed completed, it must be signed off by all team members. In the event of a member not being present during a code review, the code review will either be pushed back to a further date, or the code review will continue as planned, and the absent member will be required to review the code on their own time, in a timely manner.

## 6.2 Approach

All documents required by the Prof. will be baselined. In the event of a change in the document, the previous text that shall be replaced will be highlighted to ensure that the team is able to differentiate the previous version to the new one. The person editing the document will be the person in charge of making this change.

The code will be stored within a repository, where the files will have revision history, along with having the contributor’s name. The code reviews will happen during team meetings and be reviewed from the repository.

During the planning stages of the project, we will devise a roadmap with our clients require a signature from both parties to ensure that the road map will not change unless both parties agree. As previously stated, both clients will need to sign off on a change of delivery.

## 6.3 Impact

There’s a wide variety of positive and negative impacts. Some of the positive impacts is that it will hold people accountable. As a team, we will be required to stick to the delivery dates. The clients will also be accountable to give us what they want well in advance. Code reviews allow us, again, to hold people accountable. From code reviews, we will also be able to seek out potential bugs in the software, determine if there are any bad practices that we can rewrite and make sure we’re on the same page of how the code works and are able to ask questions if we are unsure.

Some negative impacts of using this are that time is a big factor within this project. We only have 8 months to totally complete this project. Creating road maps, doing code reviews along with doing version control on artifacts all cost time, which is very limited for us.

# **7.0 SQA Tools, Techniques, Methods**

We will be using a number of tools in order to make sure our software is at the highest level possible. Due to the nature of our project, we will be using web testing tools. Some of the tools we will be using include Selenium Web Driver, Cypress.io, and possibly Test.ai.

Along with testing during, we will be creating automated tests that allows us to make sure any of the effort we have done in the past is not broken due to changes in the code. On top of making sure our code stays at its peak we will also be doing a combination of white box and black box testing.

Before anyone submits code to the project, they will have completed white box testing during development and directly after completion of a task. Once the white box testing is completed, the same user will complete black box testing on the said piece of code, the programmer will make sure all of the automated tests work properly, along with creating new tests (if applicable).

# **8.0 Appendix**

N/A