Mitigation Repository

Validation Document

Prepared by Anthony Tesoriero

Team: River Otters

Date: 27 April 2020

Version: 2

Table of Contents

1. Purpose

This document demonstrates that all requirements have been met, per the requirements document, by establishing a trace between the initial requirements, through to final testing. The document defines project deliverables, and other terms that may be used by those deliverables, such as testing types in the testing plan.

2. Summary of Deliverables Generated

This section lists all written deliverables generated during the project, in order to trace the SDLC phases.

2.1. Project Initiation Document

Version: 3

Provides the foundation of the project. Specifies project importance, along with what will be delivered, as discussed with sponsors. Outlines the original project goals, and original ideas on how the team plans on working. The document also contains a risk assessment table, and several initial risks with the project. The document then states the estimated project cost, and the estimated ROI after the project requirements are completed.

2.2. Requirements Document

Version: 3

Outline project requirements as discussed with sponsors, and current plan as of writing. The document defines terms necessary to specify what is required. It also contains necessary diagrams, such as ER and Use Case Diagrams, to display what will be implemented. Specific system requirements are included to break down larger requirements into atomic sub-requirements. The document also states how the project will evolve overtime.

2.3. Design Document

Version: 2

Describes all architecture and design decisions that will be implemented in the project. The document is made up of several diagrams used to depict specific aspects of the project. There are also visuals of the UI. Diagrams, such as the use case diagram, break down the original diagrams found in the requirements document. The document also states what tools have been and/or will be used, and their purpose.

2.4. Test Scripts and Validation Procedures

Version: 1

Contains what and how the project will be tested, including project test scripts. Each test has a traceability matrix to show what is being done, and the outcomes. The document uses several testing methods to test different aspects of the project in different ways. The document also contains a validation strategy and methodology, to break down what is being done.

2.5. Validation Document

Version: 1

See Section 1.

2.6. Implementation Document

Version: 1

To be delivered.

3. Validation Plan

These tests are done throughout the project, and using the test scripts (Section 2.4).

3.1. Unit Testing

Testing a small piece of code to see if the result is as expected. This type of testing was done throughout the project, and continues to be tested during the testing phase. This is very important to test crucial features that are necessary for the project to work correctly.

3.2. Integration Testing

Making sure different pieces of the code and different segments work together as they should. The team tests a very wide scope to cover each part of the project, and make sure things are working correctly. One big part of this is making sure each part of our front-end is working with its corresponding back-end piece.

3.3. System Testing

After unit and integration testing is complete, there is system testing, to make sure the project as a whole is working. This includes both functional and non-function testing of the project, and making sure everything works especially from a user perspective.

3.4. IQ

Verifying that everything needed is installed correctly, and works properly. This is very important for us on the back-end in the AWS database, making sure that all external software we use is doing what it is supposed to so the project can function correctly.

3.5. OQ

Testing functional requirements of the project to make sure all requirements are working as they should. This testing makes sure that all of our key functionality is working, and our project is actually usable by the customer.

4. Operations and Maintenance Procedures

As time goes on, external parts of the project, such as the database and installed software, may become outdated. In this case, if needed, the software should be updated. The project should then be tested using the old testing scripts, and any new tests that may need to be done due to the new software. Should old tests fail, these parts of the project should be fixed, and the tests should be run again. All changes should be documented in a new testing script version.

5. Roles and Responsibilities

5.1. Roles

The team consists of the following people performing the following roles:

Name	Role
Alyssa Indriso	Scrum Master
Anthony Tesoriero	Product Owner
David Glennan	Development Team
Kristen Stansfield	Development Team
Michael Burke	Development Team
Theresa Morris	Development Team

5.2. Responsibilities

The following roles perform the following responsibilities:

Role	Responsibilities
Scrum Master	Provide guidance to work efficiently through sprints. Work with team to decide what PBI's will be done during the sprint. Also observes the definition of done.
Product Owner	Communicate between the scrum master and the sponsor to identify project requirements. Assign priorities to PBI's to help effectively divide work.
Development Team	A group of developers working on project software development. Work to have proper increments for every sprint. May split into sub-teams to increase effectiveness and efficiency.

6. Terms and Definitions

6.1. Acronyms

Acronym	Definition
AWS	Amazon Web Server
ER	Entity Relationship
IQ	Installation Qualification
OQ	Operational Qualification
PBI	Product Backlog Item
ROI	Return on Investment
SDLC	Software Development Life Cycle
UI	User Interface

6.2. Terms

Entity Relationship Diagram: a diagram to visualize data relationship and dependencies between entities inside of a database.

Forking: the ability to clone a mitigation so users can work on mitigations to add different solutions to the same mitigation.

Repository: a central table for all storage locations. It is used to implement version control and can store multiple versions of a data record. There are countless types of mitigations in cybersecurity so using centralized data, it ensures that all of the data is being pulled and stored in one location.

Risk Mitigation: a strategy to prepare or decrease the presence of threats faced by data center. Usually illustrated in steps for a user to follow to reduce possible negative effects on a system's data center.

Security Control: a way to reduce or mitigate a risk to assets (ex: physical property, information data, hardware systems). These are classified by multiple criteria which is given a category (ex: directive, preventative, detective, etc.) and a type (ex: physical, administrative, or technical)

Use Case Diagram: a diagram to visualize the different roles throughout the system and how those roles interact with the system.

7. Revision History

Rev. #	Edited By	Reason	Date
1.0	Anthony Tesoriero	Initial Version	20 April 2020
2.0	Anthony Tesoriero	Version 2	1 May 2020