SE 3XA3: Test Plan
Q-aRt Code

Team 05, Q-aRt QTs
Elton Schiott schiotek
Emilio Hajj hajje
Liam Duncan duncanla

October 21, 2016

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

1 General Information

1.1 Purpose

This document is an outline of the procedures intended to be used to verify that the software project currently under development, Q-aRt Code satisfies requirements as specified in the SRS document. This plan and the majority of the tests described within have been constructed prior to the completion of the project implementation. As such, the intention of the document is to be used as a reference during the implementation of the software, and during testing.

1.2 Scope

The majority of the project is encoding binary strings from the input data, therefore, the scope of the testing for this software will largely be ensuring correctness of these generated strings, but also includes readability of the final generated QR code, as well as image quality.

1.3 Acronyms, Abbreviations, and Symbols

Table 2: **Table of Abbreviations**

Abbreviation	**Definition**
Abbreviation1	Definition1
Abbreviation2	Definition2

Table 3: **Table of Definitions**

Term	**Definition**
Term1	Definition1
Term2	Definition2

1.4 Overview of Document

2 Plan

2.1 Software Description

The software under development for this project, Q-aRt Code, is a QR code generator that has the added functionality of creating artistic QR codes. The modular decomposition of the software is as follows:

- Encode binary strings from the input data .
- Generate error correction codewords from the binary strings.
- Structure data and error correction binary strings, including interleaving.
- Create a matrix using the generated data according to QR code standards.
- Combine generated QR Code with input image to create artistic QR code.

2.2 Test Team

Each member of the test team will be responsible for both the creation of tests and their execution. The members of the test team are as follows:

- Elton Schiott
- Emilio Hajj
- Liam Duncan

2.3 Automated Testing Approach

2.4 Testing Tools

Unit testing will be conducted using the Python unittest unit testing framework.

2.5 Testing Schedule

See Gantt Chart at the following url ...

3 System Test Description

3.1 Tests for Functional Requirements

3.1.1 Area of Testing1

Title for Test

1. test-id1

   Type: Functional, Dynamic, Manual, Static etc.
   Initial State:
   Input:
   Output:
   How test will be performed:

2. test-id2

   Type: Functional, Dynamic, Manual, Static etc.
   Initial State:
   Input:
   Output:
   How test will be performed:

3.1.2 Area of Testing2

...

3.2 Tests for Nonfunctional Requirements

3.2.1 Area of Testing1

Title for Test

1. test-id1

   Type:
   Initial State:
   Input/Condition:
   Output/Result:
   How test will be performed:

2. test-id2

   Type: Functional, Dynamic, Manual, Static etc.
   Initial State:
   Input:
   Output:
   How test will be performed:

3.2.2 Area of Testing2

...

4 Tests for Proof of Concept

4.1 Area of Testing1

Title for Test

1. test-id1

   Type: Functional, Dynamic, Manual, Static etc.
   Initial State:

Input:

Output:

How test will be performed:

2. test-id2

   Type: Functional, Dynamic, Manual, Static etc.

   Initial State:

   Input:

   Output:

   How test will be performed:

4.2 Area of Testing2

...

5 Comparison to Existing Implementation

6 Unit Testing Plan

6.1 Unit testing of internal functions

6.2 Unit testing of output files

References

7 Appendix

This is where you can place additional information.

7.1 Symbolic Parameters

The definition of the test cases will call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.

7.2 Usability Survey Questions?

This is a section that would be appropriate for some teams.