|  |
| --- |
| UAHealth Bit Vault Software Test Plan |
| CPE 656/658 Software Studio |
| Timothy R. Wilkins  Whit J. Sisulak  Glen L. Riden  James J. Duggan IV |

10/12/2015

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision # | Revision Date | Description of Change | Author |
| 0.1 | 10/11/15 | Initial Draft | T. Wilkins |
| 0.2 | 10/12/15 | Added cover page, revision history, headers, and footers. | J. Duggan |
| 0.3 | 10/13/15 | Added section 4.2.5 | T. Wilkins |
| 0.4 | 10/20/15 | Refactored to reflect design change. Added up to section 4.2.3 | T. Wilkins |
| 0.5 | 10/25/15 | Added up to section 5.1.2 | T. Wilkins |
|  |  |  |  |

Table of Contents

[Revision History i](#_Toc432432708)

[1 iii](#_Toc432432709)

[2 Introduction 1](#_Toc432432712)

[2.1 Objectives 1](#_Toc432432713)

[3 Scope 1](#_Toc432432714)

[3.1 Identification 1](#_Toc432432715)

[3.2 System Overview 1](#_Toc432432716)

[3.3 Document Overview 2](#_Toc432432717)

[3.4 Relationships 2](#_Toc432432718)

[4 References 2](#_Toc432432719)

[5 Software Test Environment 2](#_Toc432432720)

[5.1 Data Collection Test Environment 3](#_Toc432432721)

[5.1.1 Software 3](#_Toc432432722)

[5.1.2 Hardware and Firmware 3](#_Toc432432723)

[5.1.3 Other Materials 3](#_Toc432432724)

[5.1.4 Proprietary Nature, Acquirer’s Rights, and Licensing 4](#_Toc432432725)

[5.1.5 Installation, Testing, and Control 4](#_Toc432432726)

[5.1.6 Participating Organizations 4](#_Toc432432727)

[5.1.7 Personnel 4](#_Toc432432728)

[5.1.8 Orientation Plan 5](#_Toc432432729)

[5.1.9 Tests to be Performed 5](#_Toc432432730)

[5.2 Data Analysis Test Environment 5](#_Toc432432731)

[5.2.1 Software 5](#_Toc432432732)

[5.2.2 Hardware and Firmware 5](#_Toc432432733)

[5.2.3 Other Material 5](#_Toc432432734)

[5.2.4 Proprietary Nature, Acquirer’s Rights, and Licensing 5](#_Toc432432735)

[5.2.5 Installation, Testing, and Control 5](#_Toc432432736)

[6 Test Identification 6](#_Toc432432737)

[6.1 General Information 6](#_Toc432432738)

[6.1.1 Test Levels 6](#_Toc432432739)

[6.1.2 Test Classes 6](#_Toc432432740)

[6.1.3 General Test Conditions 6](#_Toc432432741)

[6.1.4 Test Progression 6](#_Toc432432742)

[6.1.5 Data Recording, Reduction, and Analysis 6](#_Toc432432743)

[6.2 Planned Tests 6](#_Toc432432744)

[6.2.1 Items to be Tested 6](#_Toc432432745)

[6.2.2 Project-unique Identifier of a Test 6](#_Toc432432746)

[7 Test Schedule 6](#_Toc432432747)

[8 Requirements Traceability 6](#_Toc432432748)

[9 Notes 6](#_Toc432432749)

[10 Annexes 6](#_Toc432432750)

[11 Risks 7](#_Toc432432751)

[11.1 Schedule 7](#_Toc432432752)

[11.2 Technical 7](#_Toc432432753)

[11.3 Management 7](#_Toc432432754)

[11.4 Personnel 7](#_Toc432432755)

[11.5 Requirements 7](#_Toc432432756)

[12 Approvals 7](#_Toc432432757)

Software Test Plan

# Introduction

This document shall be used to create a test plan for the medical project.

## Objectives

The main objective of this document is to come up with all the necessary parts of a fully-fledged test plan. This includes identifying the scope of testing that will be done, identifying any references needed to perform the testing, defining the environment in which the tests will be run, and identifying the classifications of the tests themselves. This document will also explore any risks involved with meeting the test plan. Approvals will be noted at the end of the document.

# Scope

## Identification

This test plan covers the testing of two key pieces of software, the data collection software and the data analysis software. All testing outlined in this document must be run against the final version of these software. The final version of the application containing both pieces of software that passes all testing shall be known as version UAHealth Bit Vault 1.0.

## System Overview

Below is a linked document containing the system overview.



## Document Overview

This document contains information relevant only to the software, testing, and planning associated with such. None of the material in this document is to contain actual names, medical data, or information about actual people living or dead. Medical information is private and no actual data from actual people will be included in this document. As such, this document may be distributed freely and made publicly available.

## Relationships

This document is closely related to the Software Development Plan (SDP). As the SDP is created, this document will need to be maintained to keep its relevance. Any tests described by this document will be closely related to the features, classes, or functions described in the SDP and thus as these elements are updated or changed, this document will need to be updated or changed. Updates to this document will be under revision control as described in the Configuration Management Plan.

# References

# Software Test Environment

This section describes the test environment. The environment includes the software and hardware needed to test the product. This section will also cover the installation, testing, and control processes of the test software and hardware.

## Local Testing Environment

This is the environment local to the developer’s machine. Testing here will include mostly unit testing, but may also include some component or end-to-end testing.

### Software

Tools to be used during testing in the local environment are listed below. A detailed description of the tool, its intended use, where or who to get it from and any other relevant details are below. These will be added to as we develop the design. As the design changes new or different tools may become necessary.

* Visual Studio Enterprise 2015 – This tool will be utilized for local development of the software. Visual Studio Enterprise 2015 and the unit test plugin can be obtained here: [http://e5.onthehub.com/d.ashx?s=vx43ohwn8y](http://e5.onthehub.com/d.ashx?s=vx43ohwn8y" \t "_blank)
* Windows Server 2008 – This will be the developing OS. All development and testing should be performed on this OS for this piece of software. It can be downloaded here: [http://e5.onthehub.com/d.ashx?s=vx43ohwn8y](http://e5.onthehub.com/d.ashx?s=vx43ohwn8y" \t "_blank)
* GitHub Desktop – This is the repository software to be used to store all tests. GitHub is located at [www.github.com](http://www.github.com).
* NUnit 2.6.4 – This will be used for unit test development. It can be downloaded here: [www.nunit.org/index.php?p=download](http://www.nunit.org/index.php?p=download)
* Mock Classes – A mock of each class necessary for performing unit tests will need to be created. It will be the responsibility of the team members to create these mock classes.

### Hardware and Firmware

Hardware to be used during testing in the local environment is listed below. A detailed description of the hardware, its intended use, where or who to get it from and any other relevant details are below.

* PC – a PC will be required for testing. This will be the responsibility each team member to provide for himself.
* Internet Connection – an internet connection will be required for saving and retrieving the tests. It is also the responsibility of each team member to provide this for himself.

### Other Materials

Any other materials to be used during testing on the local environment are listed below. A detailed description of the material, its intended use, where or who to get it from and any other relevant details are below.

* Mock patient data – This is made up data that can be imbedded in the tests for automation purposes. This is to be provided by the team.

### Proprietary Nature, Acquirer’s Rights, and Licensing

### Installation, Testing, and Control

Each team member shall be responsible for downloading and installing Visual Studio Enterprise 2015 and Windows Server 2008. Code under test and the tests themselves can be downloaded off of GitHub. In order to download from GitHub, the GitHub Desktop software needs to be installed. Follow this tutorial to perform the installation: <https://help.github.com/desktop/guides/getting-started/>

All the mocks will need to be developed. There should be a mock of all the classes needed in order to get the minimum required code and path coverage into unit tests. These mocks should be developed at the same time or before the classes they are needed to test. Following the CM plan for software development performs testing of a mock.

A PC is required to be provided by each team member for testing. Each team member will also need an Internet connection in order to download the tests and code under test.

GitHub and the Internet connection are under test and control by the provider. Windows Server 2008 and Visual Studio are under test by third parties and the versions are under control by the configuration management plan. The PC is assumed to be working and maintained by the team member.

### Participating Organizations

Organizations involved in the testing of the data collection software are listed below. Their roles are described with each role.

* Med656 Team – The Med656 team will be responsible for writing unit tests, acceptance tests, and running both of them.
* Customer – The customer will be responsible for customer acceptance tests.

### Orientation Plan

All team members are to go through the following tutorial for training purposes for GitHub: <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>. There will also be a training session for C#/Visual Studio let by James Duggan.

### Tests To Be Performed

## Staged Testing Environment

This will be the testing environment on our controlled UAH server. Testing here will be focused on end-to-end tests but may also cover some component testing.

### Software

Tools to be used during testing in the staged environment are listed below. A detailed description of the tool, its intended use, where or who to get it from and any other relevant details are below. These will be added to as we develop the design. As the design changes new or different tools may become necessary.

* Windows Server 2008 – This will be the OS running on the stage server. The OS on the server should be running on top of a VM. The customer should provide the OS and the VM.
* Visual Studio Enterprise 2015 – This will be used to develop the tests for this environment.
* Selenium 2.48.0 for C# – This will be used to execute the end-to-end tests. It can be obtained here: <http://www.seleniumhq.org/download/>.

### Hardware and Firmware

Hardware to be used during testing in the staged environment is listed below. A detailed description of the hardware, its intended use, where or who to get it from and any other relevant details are below.

* Server – There will be server hardware provided by the customer. The maintenance and control of this hardware is up to the customer.
* Hypervisor – There will be a hypervisor installed on the server of the customer’s choice. Maintenance and control of the host OS and hypervisor are up to the customer.
* VPN Access – The customer will provide VPN access to the guest OS. This will be up to the customer to provide and the team to maintain.

### Installation Testing and Control

The Server, hypervisor, guest OS, and VPN access shall be installed by the customer. The Server, hypervisor, and VPN access shall be tested and controlled by the customer. Issues with these three components will be brought to the customer for resolution. The guest OS shall be tested and controlled by the team.

Visual Studio and Selenium shall be installed by the team onto the guest OS. Issues with these components will be resolved by the team.

### Participating Organizations

Organizations involved in the testing of the data collection software are listed below. Their roles are described with each role.

* Team – The team will be involved in writing and performing tests. The team will also be in charge of installing, configuring, and maintaining Visual Studio Enterprise 2015 and Selenium 2.48.0.
* Customer – The customer will be involved with reviewing this document, test case coverage, and test results. The customer will be responsible for signing off on this document. The customer will also be responsible for signing off on the test results of the final release of the software.

# Test Identification

This section identifies all the tests that will apply to this software test plan. It gives a description of each test. Each test number can be used to identify which requirement it belongs to in the test traceability matrix.

## General Information

This section describes general information about the tests in this test plan. The information includes test levels, test classes, test conditions, test progression, and data recording/reduction/analysis.

### Test Levels

The following is a list of levels at which the software will be tested.

* Procedure level – This level will be covered by a minimum of one unit test per procedure. This shall apply to both class procedures as well as any global procedures.
* Class level – This level will be covered by acceptance tests. Each of these tests will be written to directly or indirectly, partially or fully test the software with regard to functionality or non-functional requirements.
* UI level – This level will be covered by end-to-end tests. These tests will be developed directly in relation to a functional or non-functional requirement.

### Test Classes

The following is a list of classes/types of tests that will be performed.

* Functional – These will test the behavior of a specific procedure/class/feature with respect to its intended behavior.
* Timing – Tests of this category will check the responsiveness of the system at key points in the software.
* Scale – Tests of this category will check the capacity of the system at key points in the software.
* Erroneous Input – Tests of this category will check the ability of the system to handle invalid input.
* Service Interruption – Tests of this category will check the ability of the system to recover from a loss of some resource (power, network, database…).
* Installation/Setup – Tests of this category will check the ability of the functionality of the setup and installation scripts.
* Security – Tests of this category will check the ability of the system to limit users to their respective features.

### General Test Conditions

The following conditions will apply to all tests that are run in this test plan:

* All tests shall pass a minimum of three consecutive test runs for their results to be marked as passed in the customer acceptance test report.
* Before code is committed to each team member’s development branch, all unit tests must be passing.
* Unit tests shall maintain a minimum of 70% code coverage. Unit test coverage is the first chance to uncover bugs and helps code stability, but requiring 100% code coverage without development team buy in can cause a lac of quality in unit tests. Therefore the 70% was selected to encourage high quality unit test development.
* Acceptance tests shall cover 100% of the functional requirements. Properly written acceptance tests are one of the best ways to regularly check that the product being developed will match what the customer wants. Therefore 100% code coverage has been required to verify that all the customers’ requests are met.

There shall be three categories of tests: CI tests, customer acceptance tests, and regression tests. CI tests shall consist of very fast tests. All tests in this category must complete running within 5 min. This helps to guarantee that each developer will run the test before or immediately after commitment of any changes. Tests that will push the test run time beyond 5 min. will be put into the customer acceptance category. These tests will be run before code is released to the customer at a minimum. The customer acceptance category shall be subdivided into functional categories that can be run independent of each other to test specific features when hotfixes have been applied. Customer acceptance tests will include tests for current features. Any tests related to outdated, deprecated, or existing features that are no longer being developed should be moved to the regression category. The regression category tests shall be run before each major point release.

### Test Progression

N/A

### Data Recording, Reduction, and Analysis

Unit test results shall not be saved for evaluation by the customer. Acceptance test results and regression test results shall be saved for evaluation. All test results and test artifacts generated during the execution of automatic testing of acceptance or regression tests shall be zipped together and delivered to the customer. Test results of semi-automatic or manual tests shall be filled into an appropriate spread sheet. Any failures in all test results delivered to the customer shall be documented and brought to the attention of the customer.

## Planned Tests

### Data Collection Software

#### User Login Customer Acceptance Tests

1. Objective – These tests will verify the requirements in section 3.1.1 of the Software Requirements document have been met.
2. Test Level – These tests will be at the UI level.
3. Test Class – These tests will include classes: Functional, Scale, Erroneous Input, and Security.
4. Qualification method - ???
5. Requirements - See Requirements Traceability.
6. Special requirements – N/A.
7. Recorded data type – automated test generated artifacts, test results.
8. Type of data recording/reduction/analysis – Logger (Info and Debug), test framework test report.
9. Assumptions/Constraints – N/A.
10. Safety, security, privacy considerations – Tests related to security must pass with no exceptions.

### Data Analysis Software

### Project-unique Identifier of a Test

# Test Schedule

# Requirements Traceability

# Notes

# Annexes

# Risks

## Schedule

## Technical

## Management

## Personnel

## Requirements

# Approvals