*Florida International University*

*School of Computing and Information Sciences*

CIS 4911 Senior Capstone Project

Test Case Management System

Early Draft Requirements Document (RD)

September 2014

**Group Members**

Justin Phillips  
Karina Harfouche

**Mentors**Tariq King  
Dionny Santiago

**Instructors**Masoud Sadjadi  
Juan Caraballo

**Requirements Document (RD)**

**The bullets represent the sections before chapter 1 “Introduction”, numbers on the left represent the corresponding chapters and sections.**

· Copyright and trademark notices, restrictions on copying or distributing the documentation, information for contacting the issuing organization (reader’s comments), warranties, contractual obligations or disclaimers, and general warnings and cautions.

·

**Abstract**

The present requirements document consists of an introduction and overview of the system along with the detailed requirements discussed with the Test Case Management system client. The requirements will provide a general picture in order to create the necessary use cases, UML, diagrams and overall design of said functional and non-functional requirements specified by the software testers’ representatives of Ultimate Software. A requirements study describing the relationships and dependencies of each feature and functionality will cover all aspects of the current solution the members of this project along with Ultimate Software have proposed to manage test case suites and automation. The depiction of clear definitions and UML diagrams of Ultimate Software’s requirements will be an essential guide of this document towards the implementation of a secure and user friendly application. The elaborated solutions as well as the studied alternatives presented are focused on improving the overall tester experience of documenting and managing all the tasks related to software testing at Ultimate Software. The human, hardware, software and other resources estimated on the planning section of this deliverable are in proportion to the system’s functional and nonfunctional requirements.

Table of Contents

**1. Introduction**

The current chapter presents a general overview of the Test Case and Automation Management software. Initially, the problem definition of the project will be depicted in order to provide a context that leads to the explanation, purpose and scope of the proposed system. Prominent definitions, acronyms and abbreviations used in this document are defined in section 1.3. In conclusion, the final section will encapsulate a complete overview and main points discussed in this document.

**1.1. Problem Definition.**

Software testing is a core decision stage within a functional software’s development and operation life cycle, testing determines the correctness, completeness, and overall compliance with a client’s requested features and requirements. Companies such as Ultimate Software depend heavily on efficient and practical management of software testing in order to confidently validate their software products with their customer requirements. While Ultimate Software’s testing team manages large sets of test cases using Microsoft Test Manager (MTM) in conjunction with Microsoft’s Team Foundation Server (TFS), several shortcomings have been found in MTM . The current software testers’ experience when creating, editing and managing the tasks related to testing in general could improve in order to significantly and directly enhance the time it takes to fully test software as well as the overall assertiveness and ease of spotting blocking issues that ultimately affect the customer’s satisfaction within a given software product.

**1.2. Scope of system.**

The proposed Test Management system’s requirements and development will be solely based and personalized on the needs and specifications of the Solution Testing Team at Ultimate Software. As the requirements and use case sections of this document specifies, the scope of the system includes the general tasks involved with the creation, edit, update and deletion of test case documents coupled with the resulting database modifications done from the application as well as the currently used TSF and MTM dependencies. The automation aspect of the system will be pursued only if permissible by the time constraints and resources for this project. In the case that time constraints impede the full development of the application’s ability to automate test cases, analysis and proper planning will still be made accordingly in order to develop an application that allows the addition of an automation component.

**1.3. Terminology - Definitions, acronyms, and abbreviations.**

**COCOMO:** Constructive Cost Model

**MTM:** Microsoft Test Manager

**SCM:** Source Code Management

**SW**: Software

**TFS:** Team Foundation Server

**WA:** Work Activity

**1.4. Overview of document – brief explanation of what to expect in chapters 2 through 6**

This document incorporates the early definitions of the project necessary to complete an accurate requirements analysis for the development of the Test Case Management System. The primary objective of this document is to present the project’s overview found on this chapter as well as an in depth analysis of the system requirements. The second chapter of this document provides an overview of the current system. Within chapter 3, the software project’s development and management organization are described in detail. Chapter 4 describes the system requirement analysis and supporting artifacts. Chapter 5 contains the glossary of project related terms. To conclude the appendix provides many of the initial project artifacts which support the system requirements analysis.

**2. Current System**

The current system, within the scope of this project, is centered around MTM utilized in conjunction with TFS. These applications and services are used to create test suites, cases, and steps in addition to providing autonomous build, test, and deployment services; as well as reporting services. Also, MTM supports manual testing via reporting and rich media capturing capabilities. The system provides for the ability to then query reports and tests.

The most major limitation of the current system is the usability governing creation or cloning of test suites, cases, and steps. The usability limitations exist within the structural limitations of test cases, drill-down navigation, and the complexity/limited feature set of the user interface. The client would like the ability to create two levels of test cases, one that is high level and one at an implementation details level; which the current system does not support. MTM provides a drill- down navigation of test suites and cases that allow for attachments to be added. However, viewing these attachments using the drill-down navigation creates for an arduous process. Lastly, in many areas test case creation is too complex, with too busy of a user interface while also not providing necessary features such as copying of cases and steps.

Another limitation of the current system is that MTM cannot associate test suites, cases, or steps with the code repository. Thus, there does not exist a method for which the tester can be notified about file changes associated with a given test element.

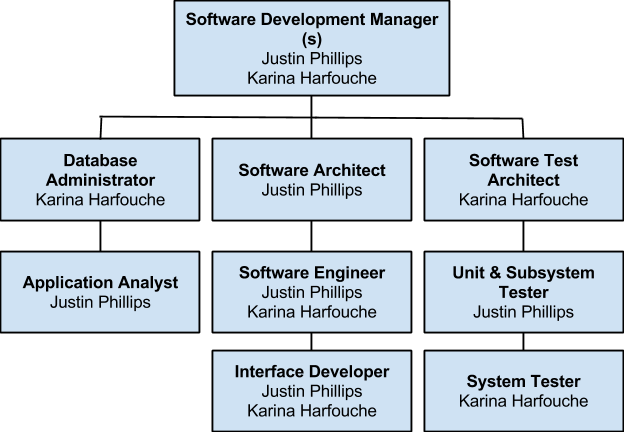
Finally, MTM does not provide a mechanism in which the results of automated test may be displayed. This provides for a fragmented testing system which requires multiple applications to determine the state of the tests and displaying test associations.

**3. Project Plan (This deliverable only)**

The project organization section focuses on the visualization and analysis of the general software organization as well as the software support and service organization. A hierarchical chart will be presented in order to illustrate the proposed structure for the Test Case Management System’s working personnel

**3.1. Project organization**

This is the software development organization for the Test Case Management System:



**3.2. Work breakdown – identification of milestones and deliverables (refer to project schedule in Appendix A and the diary in appendix B).**

The following table identifies the major tasks, milestones and deliverables along with each respective duration dates:

|  |  |  |  |
| --- | --- | --- | --- |
| **Task Name** | **Start Date** | **End Date** | **Duration (days)** |
| **Requirement Elicitation** | **09/04/14** | **09/15/14** | **8** |
| **Meet with client** | **09/04/14** | **09/04/14** | **1** |
| **Determine Requirements** | **09/04/14** | **09/15/14** | **8** |
| **System Analysis** | **09/05/14** | **09/15/14** | **7** |
| **Determine System Feasibility** | **09/05/14** | **09/10/14** | **4** |
| **Compare with Alternatives** | **09/10/14** | **09/15/14** | **4** |
| **Present Deliverable** | **09/15/14** | **09/15/14** | **1** |
| **Software Project Planning** | **09/26/14** | **09/30/14** | **3** |
| **Scheduling** | **09/26/14** | **09/26/14** | **1** |
| **Resource Allocation** | **09/27/14** | **09/27/14** | **1** |
| **Risk Plan** | **09/27/14** | **09/30/14** | **3** |
| **Software Project Organization** | **09/29/14** | **10/06/14** | **6** |
| **Personnel Structurization** | **09/29/14** | **09/29/14** | **1** |
| **Obtaining Tools** | **10/01/14** | **10/01/14** | **1** |
| **Define Metrics** | **10/02/14** | **10/03/14** | **2** |
| **Acceptance of Project Organization** | **10/04/14** | **10/06/14** | **2** |
| **System Design** | **10/06/14** | **10/10/14** | **5** |
| **Create Use Cases** | **10/06/14** | **10/06/14** | **1** |
| **Subsystem Decomposition** | **10/07/14** | **10/07/14** | **1** |
| **Determine Security and Privacy** | **10/08/14** | **10/08/14** | **1** |
| **Create DB Schema** | **10/09/14** | **10/09/14** | **1** |
| **Hardware & Software Mapping** | **10/10/14** | **10/10/14** | **1** |
| **Object Design** | **10/12/14** | **10/24/14** | **11** |
| **Create Class Diagram** | **10/12/14** | **10/12/14** | **1** |
| **Create OCL statements** | **10/13/14** | **10/13/14** | **1** |
| **Create Sequence Diagram** | **10/24/14** | **10/24/14** | **1** |
| **System Implementation** | **10/10/14** | **11/13/14** | **25** |
| **Create/Configure Database** | **10/10/14** | **10/14/14** | **3** |
| **Implement User Interface** | **10/14/14** | **10/24/14** | **9** |
| **Implement System Logic** | **10/25/14** | **11/13/14** | **15** |
| **Present Deliverable** | **10/28/14** | **10/29/14** | **2** |
| **Testing** | **11/13/14** | **12/02/14** | **14** |
| **Create & Implement Test Cases** | **11/13/14** | **11/17/14** | **3** |
| **Run test suites** | **11/13/14** | **11/19/14** | **5** |
| **Evaluate & Fix Defects** | **11/19/14** | **12/02/14** | **10** |
| **Present Final Deliverable** | **12/15/14** | **12/16/14** | **2** |
| **Present Deliverable to Client** | **12/15/14** | **12/15/14** | **1** |
| **System Release** | **12/16/14** | **12/16/14** | **1** |

**3.3. Cost Estimate – cost to develop the software system.**

**4. Proposed System Requirements**

The new system shall provide the client with the ability to create test suites, cases, and steps efficiently. The system shall interface with TFS and be configurable for both TFS communication and code repository linking. Furthermore, the system shall allow test elements to be associated with a code repository. The system shall also support adding attachments to test elements. This entire system shall be secured behind a single sign-on login which is linked to domain accounts. Each feature within the proposed system shall provide a easy to use, simple interface to the system actors.

**4.1. Functional Requirements**

The implementation of the Test Management System will include ten main functionalities that the system shall perform:

1. The system shall allow for testers within the configured Windows domain to have single sign-on access using the account the tester is signed into a Windows machine with.
2. The system shall provide a mechanism to configured the test management backend system.
3. The system shall provide an interface for creation of test suites, cases, and steps with fluid, intuitive authoring experience.
4. The system shall provide testers with the ability to add attachments and associate them with test suites, cases, and steps. The attachments should then be viewable with ease and with as few clicks as possible.
5. The system shall provide a mechanism to associate test suites, cases, and steps with a code repository or automation.
6. The system shall provide an interface to configure the interaction with a code repository.
7. The system shall display test suites, cases, and steps in a clean explorer or tree view throughout the user navigation within the system.
8. The system shall provide a means to display data categorically, organically, and in a logical view.
9. The system shall provide a simple search interface for searching the test suites, cases, and steps.
10. The system shall provide a logout mechanism so that the current user may be logged out and a user other than the user logged into the current Windows machine may login to the system.

For each functional requirement state the associated non-functional requirements, if any, for *Usability, Reliability, Performance,* and *Supportability*.

**4.2. Analysis of System Requirements**

Analysis models – contains the complete functional specification and is mainly for the designers and programmers. This section describes the diagrams in the Appendices B - D and validates the models against the use cases.

**4.2.1. Scenarios**

**4.2.2. Use case model**

**4.2.3. Static model e.g., object diagrams, class diagram**

**4.2.4. Dynamic model e.g., sequence diagrams or state machines**

**5. Glossary - define terms used in document, especially domain specific terms.**

**6. Appendix**

**6.1. Appendix A - Complete use cases**

**6.2. Appendix B - Use case diagram using UML**

**6.3. Appendix C - Static UML diagram**

**6.4. Appendix D - Dynamic UML diagrams**

**6.5. Appendix E - User Interface designs.**

**6.6. Appendix F - Diary of meeting and tasks.**

**Diary Entry 1:**

**Date:** Thursday, September 5, 2014

**Location:** ECS Lab 212

**Start time**: 3:00 pm

**End time:** 4:35 pm

**In Attendance:** Dionny Santiago, Tariq King, Justin Phillips, Karina Harfouche

**Late:** N/A

**7. References**

***Please email me the UML diagram in one file before the presentation.***