A NEW SOFTWARE PROJECT MANAGEMENT TOOL

A Thesis
Presented to the
Faculty of
San Diego State University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by
Sunanda Komaragiri
Summer 2012
The Undersigned Faculty Committee Approves the

Thesis of Sunanda Komaragiri:

A New Software Project Management Tool

Joseph Lewis, Chair
Department of Computer Science

Carl Eckberg
Department of Computer Science

Joseph M. Mahaffy
Department of Mathematics and Statistics

5/9/2012
Approval Date
ABSTRACT OF THE THESIS

A New Software Project Management Tool
by
Sunanda Komaragiri
Master of Science in Computer Science
San Diego State University, 2012

Software project management requires the project manager (here after, PM) to be able to manage project scope and resources within a certain time (schedule) and budget (cost constraints). A software project requires interaction between multiple users within the software organization and this is possible through the use of different project management tools. Apart from the leadership qualities of the PM, a project management tool also contributes heavily to the success of the project.

The lack of a common user interface in the existing web based project management tools leads to issues like time overhead, lack of instant resource allocation and lack of cross compatibility on multiple browsers. A survey was undertaken to account for the multiple user and common user interface software project management tools in the market. The need for a new project management tool stems from the observation that the existing common user interface project management tools are expensive and have limited features. Therefore it is hypothesized that the ability to create a new project management tool with a common user interface should contribute substantially to instant resource allocation by providing an efficient search capability mechanism to the PM.

For this project, a new project management tool was designed, tested and evaluated for four scenarios (defect capture, defect management, project status management and security). It is concluded that the new project management tool is an efficient, low cost tool that is compatible over a wide range of browsers.
TABLE OF CONTENTS

ABSTRACT ............................................................................................................................. iv
LIST OF TABLES .................................................................................................................. vii
LIST OF FIGURES ............................................................................................................... viii
ACKNOWLEDGEMENTS ....................................................................................................... x
CHAPTER

1 GENERAL INTRODUCTION ......................................................................................1
  1.1 Rationale ............................................................................................................1
  1.2 Common User Interface Concept .......................................................................2
    1.2.1 Versioning ................................................................................................ 2
    1.2.2 Unit Testing ............................................................................................. 3
    1.2.3 Refactoring ............................................................................................... 3
    1.2.4 Task Scheduling ....................................................................................... 3
    1.2.5 Log Trouble Reports into a Database and Claimed Fixes ....................... 3
    1.2.6 Keeping Track of Project Personnel and Their Assignments ............... 3
    1.2.7 Documentation about Project Code ......................................................... 3

2 RELEVANT LITERATURE AND SURVEY ..............................................................4
  2.1 Introduction ........................................................................................................4
  2.2 Software Organization Hierarchy ......................................................................4
  2.3 Software Project Management Tools .................................................................5
  2.4 Survey of Software Project Management Tools ................................................5
  2.5 Justification and Project Need ...........................................................................7

3 HYPOTHESIS ...............................................................................................................8
  3.1 Research Hypotheses .........................................................................................8
  3.2 Project Approach ...............................................................................................8
    3.2.1 Defect Capture .......................................................................................... 9
    3.2.2 Defect Management ............................................................................... 9
    3.2.3 Project Status Management ................................................................... 9
LIST OF TABLES

Table 2.1. Survey of Some of the Software Project Management Available in the Market Listed by Type of User Interface and Cost..........................................................6
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 2.1</td>
<td>Software organization hierarchy.</td>
<td>5</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>.Net framework architecture.</td>
<td>11</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>ADO.Net architecture.</td>
<td>12</td>
</tr>
<tr>
<td>Figure 5.1</td>
<td>Project status management scenario workflow.</td>
<td>18</td>
</tr>
<tr>
<td>Figure 5.2</td>
<td>Security scenario workflow.</td>
<td>19</td>
</tr>
<tr>
<td>Figure 5.3</td>
<td>Defect capture and defect management workflow.</td>
<td>20</td>
</tr>
<tr>
<td>Figure 5.4</td>
<td>Login screen.</td>
<td>21</td>
</tr>
<tr>
<td>Figure 5.5</td>
<td>Administrator main page.</td>
<td>22</td>
</tr>
<tr>
<td>Figure 5.6</td>
<td>Administration page.</td>
<td>23</td>
</tr>
<tr>
<td>Figure 5.7</td>
<td>Application configuration page.</td>
<td>25</td>
</tr>
<tr>
<td>Figure 5.8</td>
<td>Application log viewer.</td>
<td>26</td>
</tr>
<tr>
<td>Figure 5.9</td>
<td>Project – software issues.</td>
<td>27</td>
</tr>
<tr>
<td>Figure 5.10</td>
<td>Delete user.</td>
<td>29</td>
</tr>
<tr>
<td>Figure 5.11</td>
<td>Edit user.</td>
<td>30</td>
</tr>
<tr>
<td>Figure 5.12</td>
<td>Issue list.</td>
<td>31</td>
</tr>
<tr>
<td>Figure 5.13</td>
<td>Issue summary.</td>
<td>33</td>
</tr>
<tr>
<td>Figure 5.14</td>
<td>Manage roles for user.</td>
<td>34</td>
</tr>
<tr>
<td>Figure 5.15</td>
<td>New issue.</td>
<td>35</td>
</tr>
<tr>
<td>Figure 5.16</td>
<td>New project –step I.</td>
<td>37</td>
</tr>
<tr>
<td>Figure 5.17</td>
<td>New project –step II.</td>
<td>38</td>
</tr>
<tr>
<td>Figure 5.18</td>
<td>New project –step III.</td>
<td>39</td>
</tr>
<tr>
<td>Figure 5.19</td>
<td>New project –step IV.</td>
<td>40</td>
</tr>
<tr>
<td>Figure 5.20</td>
<td>New project –step V.</td>
<td>41</td>
</tr>
<tr>
<td>Figure 5.21</td>
<td>New project –step VI.</td>
<td>42</td>
</tr>
<tr>
<td>Figure 5.22</td>
<td>New project –step VII.</td>
<td>43</td>
</tr>
<tr>
<td>Figure 5.23</td>
<td>Reports.</td>
<td>44</td>
</tr>
</tbody>
</table>
Figure 5.24. Issues by status.................................................................45
Figure 5.25. Road map........................................................................46
ACKNOWLEDGEMENTS

I thank my advisor Dr. Joseph Lewis for his support, guidance and motivation during the course of this research project. I am also thankful to Dr. Carl Eckberg, Professor Joseph M. Mahaffy for their valuable suggestions, comments and for also agreeing to serve on my thesis committee.

Finally, I wish to express my gratitude to my family members who supported me during the course of my graduate study.
CHAPTER 1

GENERAL INTRODUCTION

1.1 RATIONALE

The purpose of this project is to develop an efficient and low cost project management tool to be utilized by the project managers [1] in a software organization (hereafter, firm). Per PMI (Project Management Institute), project management is defined as “the application of knowledge, skills and techniques to execute projects effectively and efficiently” [2:para. 2]. However this definition is widely used in traditional industries like construction. Per the Institute of Electrical and Electronics Engineers (IEEE), software project management is defined as the process of planning, organizing, staffing, monitoring, controlling, and leading a software project [3].

Users of project management tools in a software organization (firm) can be divided into project managers, testers, developers, quality analysts and reporters. The different groups are divided based on their roles and responsibilities within the firm. Interaction between these multiple users is commonly accomplished by utilizing different project management tools [4]. Apart from the leadership qualities of the PM, a project management tool also contributes heavily to the success of the project [3].

The lack of a common user interface [5] in the existing web based project management tools leads to issues like time overhead, lack of instant resource allocation and lack of cross compatibility on multiple browsers. In a typical scenario involving the multiple users requiring simultaneous access between a developer and a tester the following error is reported [6]. The error results in the tester locked out by the tool while trying to attempt access when a developer is already accessing the code. Access is restored only when the developer exits from the tool, resulting in a time consuming recovery process. Similarly when a project manager wants to assign inactive resources to projects, lack of efficient search capability mechanism in multiple user interface tools makes this a time consuming exercise. One other disadvantage of Multiple user interface tools like “quality center” is that it is
compatible [7] only with Internet Explorer thereby restricting its universal application to other browsers like Mozilla Firefox and Google Chrome.

Therefore the ability to create a project management tool with a common user interface should contribute substantially to instant resource allocation by providing efficient search capability mechanism to the project manager. Overall this leads to an increase in processing speed and reduction in time overhead within the firm while also maintaining the security to confidential information.

This research focuses on creating a simple web based project management tool consisting of a common user interface. Section 3.1 of this document asserts three hypotheses about such a tool.

1.2 COMMON USER INTERFACE CONCEPT

There are substantial advantages in having some project management tool features accessible to developers and other team members, rather than just the project manager. Unlike SQL, which is an interface shared by all database vendors, we do not propose an interface common to all project management tools. Rather, we refer to a common interface for the members of users (developer, tester, quality analyst, and client) apart from the Project Manager within a particular software organization. We will refer to this strategy as a “common user interface” or “multiple user tool” approach. Examples: (a) project manager only, (b) other team users. We now briefly discuss the kinds of things a project management tool should be able to do – (1) versioning, (2) unit testing, (3) refactoring facilities, (4) tasks scheduling, (5) log trouble reports into a database, (6) keeping track of project personnel and their assignments, and (7) log documentation about project code.

1.2.1 Versioning

A versioning feature allows programmers to keep track of all the revisions in the source code (i.e., new developments in the software). A couple of examples of such tools are CVS, SVN [8]. A feature such as this is normally visible to developers.
1.2.2 Unit Testing

Unit testing is performed to ensure that each unit (developed software) functions properly or as desired. Most organizations provide unit testing like junit. A feature such as this is normally visible to developers.

1.2.3 Refactoring

Refactoring is an optimization technique. It is defined as “a programming technique in which the design of the software is improved without changing its behavior” [9:para. 4].

1.2.4 Task Scheduling

Task scheduling involves creating a new project, assigning appropriate resources, defining roles and responsibilities, and estimating a project completion date.

1.2.5 Log Trouble Reports into a Database and Claimed Fixes

This is defined as the ability to display the exception details encountered within the database. This is primarily a role for the project manager.

1.2.6 Keeping Track of Project Personnel and Their Assignments

This involves managing users by tracking day by day status reports and the overall percentage of project completion. This is visible to project managers.

1.2.7 Documentation about Project Code

The history of documentation over the life of project shows the overall history of ongoing and resolved issues of a project. This can be helpful when the firm has clients that run different versions of the software.
CHAPTER 2

RELEVANT LITERATURE AND SURVEY

2.1 INTRODUCTION

This chapter presents the following information: literature pertaining to software organization hierarchy, significance of project management tools in a software organization, survey of various project management tools with different kinds of interfaces, and the rationale for using a common user interface as a model for efficient project management in this thesis. A survey of existing common user interface project management tools available in the market is also presented.

2.2 SOFTWARE ORGANIZATION HIERARCHY

Software organizations employ a broad variety of personnel in order to complete a project received from the client (in the tool presented here, client is referred as visitor). The project team consists of project managers, testers, developers, quality analysts and reporters. The following provides a brief description of the roles and responsibilities of the above noted project team members.

Project managers, also known as administratos, assign responsibilities to the rest of the project team members. The administrator has the highest level of authority over the rest of the team members present in the organization. Therefore, the PM manages the entire project from start to end. Developers code/develop the software per the project requirements set by the Project Manager. A reporter reports issues (bugs) to the developers. A tester usually gets involved with the project after the developer is done with software development. The tester will ensure that software product quality is maintained (bug free). The quality analyst ensures a better quality product. A pictorial view of this hierarchy is presented in Figure 2.1 which provides an idea of how each position fits together within a software organization.
2.3 SOFTWARE PROJECT MANAGEMENT TOOLS

An efficient software project management tool contributes to the success of any software firm. A project management tool is utilized to manage both the project activities and resources (users, users as identified in Figure 2.1). Managing large amounts of documents is a problem project managers would like to avoid and efficient project management tools are helpful. Some other important applications for which a project management tool in a software firm is utilized are tracking issues, tickets, and version control. These features are utilized by multiple users including project managers that are spread across the firm [10].

2.4 SURVEY OF SOFTWARE PROJECT MANAGEMENT TOOLS

A survey was undertaken to find out the existing software project management tools available in the market. Some of the tools utilized by software organizations are Quality Center, JIRA, Lighthouse, No Kahuna, 16 bugs, Active collab, and Microsoft Project Server [10]. Each of these tools has specific advantages and disadvantages which are briefly discussed below. Except for Quality Center, all the other listed tools have a common user interface. See Table 2.1 for survey of some of the software project management available in the market listed by type of user interface and cost [10], [11], [12].
Table 2.1. Survey of Some of the Software Project Management Available in the Market Listed by Type of User Interface and Cost

<table>
<thead>
<tr>
<th>Name of the tool</th>
<th>Type of user interface</th>
<th>Cost</th>
<th>Comments</th>
</tr>
</thead>
</table>
| Quality Center<sup>A</sup> | Multiple               | • Expensive (up to $60,000 for 10 users with an additional 15% maintenance fee) | • Complete end to-end quality management process  
 • Not very user friendly for testers  
 • Software is very slow  
 • Cumbersome search methods  
 • Not affordable for 99% of the companies |
| JIRA<sup>B</sup>           | Common                 | • $299 per month for up to 25 users  
 • Own server at $1200 to $5000 for one project | • Issue and Bug tracking software with advanced reporting, work flow mapping features |
| 16 Bugs<sup>B</sup>        | Common                 | • $15 per month                                                        | • Very simple bug tracking system  
 • Color coding system used for updates, comments and closed tickets |
| Lighthouse<sup>B</sup>     | Common                 | • $120 per month for up to 50 public projects                        | • Bug and Issue tracking application  
 • Timeline and Milestone Tracking  
 • Ticket creation for issues |
| No Kahuna<sup>B</sup>      | Common                 | • Free (very basic tool with not many features)  
 • Upgrade cost: $99 per month for 100 projects                        | • Very basic project management  
 • Issue tracking, task and activity tracking  
 • If project has more than 30 open tasks upgrade is required |
| Microsoft Project Server<sup>C</sup> | Common       | • $7,000 per license                                                 | Timesheet and project tracking tool |
| Active Collab<sup>2</sup>  | Common                 | • $199 per year  
 • Upgrade at $99 per year  
 • $199 to remove graphic Active Collab powered logo from each page | • Project Management and collaboration tool  
 • Time tracking, ticket management and milestones |

Sources:  
(A) Software Quality Assurance Forums. Cost of Quality Center License, n.d.  
(C) SoftwareMedia. Project Server 2010 - Server License and SA - Open Business, n.d.  
2.5 JUSTIFICATION AND PROJECT NEED

The importance of project management tools in the context of workflow of a software organization, and the lack of low cost and efficient common user interface project management tool with all the features needed for Project Managers, suggests a need for a new tool to satisfy the requirements.
CHAPTER 3

HYPOTHESIS

3.1 RESEARCH HYPOTHESES
The new project management tool (this thesis) was tested to verify the below three hypotheses:

- Hypothesis 1: A direct relationship exists between a common user interface and time overhead reduction.
- Hypothesis 2: A direct relationship exists between a common user interface and resource availability information (hereafter referred to as instant resource allocation).
- Hypothesis 3: A direct relationship exists between common user interface and browser cross compatibility.

3.2 PROJECT APPROACH
A new common user interface software project management tool was developed for this thesis using .NET framework on Microsoft Visual Studio Integrated Development Environment. ASP.NET, the C# language, and Microsoft SQL Server 2005 Database (to maintain information) were used to develop the code for this tool [13]. This thesis project created a new project management tool utilizing Windows forms in a simple web based Graphical User Interface (GUI). The Windows forms were also used to present information and accept input from the user along with providing a graphical display option to the user. An error validation control feature was added to the Windows forms to enhance security. As ASP.NET Web page processing occurs on the Web server, ASP.NET Web pages are compatible with any browser [14]. A Web page automatically renders the correct browser-compliant markup (XHTML or other markup language) for features such as styles and layout (hence, the developed ASP.Net with C# code is compatible on the browsers Internet Explorer, Mozilla Firefox etc. So, the cross compatibility on browsers is justified).

The new project management tool was created and tested for the following four scenarios - defect capture, defect management, project status management and security. The four scenarios covered a wide variety of issues that are encountered by all personnel within the
software organization. These four scenarios were ultimately used to verify the above hypotheses.

### 3.2.1 Defect Capture

The following options are incorporated to streamline the defect capture scenario - reporter reports bugs to the developer/PM, issues are given a deadline, clients are allowed to report defects to PM directly, and defects are assigned a unique easily traceable identification.

### 3.2.2 Defect Management

The following options are incorporated to streamline the defect management scenario - detailed description of the defects are provided in a simple web based GUI, defects are assigned by the individual to the correct team member (instead of automatic assignment by the tool as in Quality Center), detailed history of changes reported by team members over the entire project life cycle are captured, and attaching evidence files of the defects are attached.

### 3.2.3 Project Status Management

This option is intended only for use by the Project Manager. The following options are incorporated to streamline the project status management scenario – creating a new project by specifying project name, project category, assignment manager, and personnel resources along with an estimated completion date, estimated labor hours, labor spent hours and finally project progress report.

### 3.2.4 Security

The following options are incorporated to streamline the security scenario: checking for the appropriate authentications of the team members trying to access the tool and profile customization.

In Chapter 5, we discuss how these scenarios support the hypotheses noted at the beginning of this chapter.
CHAPTER 4

NEW TOOL SPECIFICATIONS


4.1 .NET FRAMEWORK

The .NET Framework is designed to simplify application development in the highly distributed environment of the Internet. It is also used to provide a consistent object-oriented programming environment encompassing all three scenarios of storage and execution of the object code (locally executed, or locally but Internet-distributed, or executed remotely). Availability of a code execution environment that minimizes software deployment and versioning conflicts, provides safe execution with other codes/applications (web based versus windows based) are some of the obvious benefits that were considered in the selection of the .NET framework for this thesis project [15].

The .NET Framework consists of a runtime environment called the common language runtime (CLR) and a set of class libraries - NET Framework class library [16]. The common language runtime (CLR) is the backbone of the .NET framework and manages the code at execution time [17]. Microsoft recommends use of class library for a wide variety of application development spanning from graphical user interface (GUI) applications to ASP.NET Web Form applications. ADO.NET is a base class library provided by .NET framework to talk to databases. Figure 4.1 [18] schematic shows the relationship between the applications (GUI) and the overall system.

4.2 ASP.NET AND C# LANGUAGES

Windows Forms is a framework for building windows client applications that utilize the common language runtime (CLR). Windows Forms applications can be written in any
language that the CLR supports including C# language. ASP.NET is used to deploy application on the internet. Data access is necessary when working with applications like GUI. A connection object establishes the connection for the application with the database. The command object provides direct execution of the command to the database. DataReader is used to provide the data from the database while the DataAdapter is used to fill the Dataset object. Dataset contains tables, columns and relationships.

4.3 MICROSOFT SQL SERVER DATABASE

Microsoft SQL Server is a relational database server whose primary function is to store and retrieve data as requested by the software applications. A schematic is shown in Figure 4.2 [19].
Figure 4.2 schematic provides a snapshot of the interaction between database and .Net framework during the application development. ADO.NET is a base class library provided by .NET framework to talk to databases.

4.4 NEW PROJECT MANAGEMENT TOOL DESCRIPTION

Data is maintained in the form of tables in the database. The tool was run for a sample of 40 projects by inputting the relevant data. This data was representative of a software organization comprising approximately 100 employees. The data used in this thesis was not obtained from an actual organization since the tools are proprietary and information confidential. Hypothetical data was created to represent a software organization and tool successfully implemented and therefore it is assumed that even with similar amount of data in the real world the tool should successfully be implemented. A normalization technique
was employed in this project to avoid any anomalies in the data [20]. As explained in the project approach in Chapter 3, the tool was tested for the four scenarios and the hypotheses were verified. Results are presented in the next chapter.
CHAPTER 5

RESULTS AND DISCUSSION

Results (in the form of screenshots) from testing and implementing the tool are presented in this chapter. The tool was tested to verify the three hypotheses, while evaluating against the four scenarios noted in Chapter 3. A brief discussion is presented below each screenshot explaining the result obtained after implementation of the model.

The reader is also introduced to the following discussion before results are presented in the next few pages in order to provide insight into the capabilities and features of the new common user interface tool that are either lacking or not available vis-à-vis in a multiple user interface tool. As already noted, users in a software organization can be divided into project managers, testers, developers, quality analysts and reporters. The users typically interact with each other utilizing project management tools. Ideally the following features are desired in any project management tool, either multiple user interface or common user interface:
(1) versioning, (2) unit testing, (3) refactoring facilities, (4) tasks scheduling, (5) log trouble reports into a database, (6) keeping track of project personnel and their assignments, and (7) log documentation about project code. The reader is referred to Chapter 1 for a detailed explanation of the above noted features. Some features like versioning, unit testing and refactoring are similar in both multiple user interface and common user interface project management tools. Hence, no further discussion is provided for the above 3 features. Instead we now provide how the remaining 4 features like task scheduling, log trouble reports into a database, keeping track of project personnel and their assignments, and log documentation about project code, are different in common user interface tools.

The following provides a brief overview of a typical application of the project management tool (multiple user interface tool versus common user interface tool) in a software organization in the context of task scheduling feature. As already noted in Chapter 1, task scheduling by a Project Manager involves creating a new project, assigning appropriate resources, defining roles and responsibilities and estimating a project completion date. A Project Manager assigning appropriate resources utilizing a common user interface
tool has instant access to the resources while the instant access feature is unavailable to the Project Manager in a multiple user interface tool. Thus, using a common user interface tool leads to reduction in time over-head compared to a multiple user interface tool. We further present screenshots in which the ability of the tool to delete and edit users, managing the roles and responsibilities are shown. The ability of the tool to modify credentials of a team member is also presented.

A project manager is responsible to manage exception details (errors) encountered in any project within the software organization. In this thesis that utilizes a common user interface tool, a new option (tab) known as “application log viewer” has been provided. By clicking this tab the project manager is instantly able to view the exceptions and manages to appropriately resolve the errors in a secured environment. In comparison, a multiple user interface tool will not provide instant access to the exceptions log that is encountered within a database (Log trouble reports into a database and claimed fixes).

A Project Manager needs to keep track of the different project personnel work loads in-order to assign new projects as and when the personnel are completed with their assignments (keeping track of project personnel and their assignments). In this thesis that utilizes a common user interface tool, a new option (tab) known as “projects” has been provided. By clicking this tab, the project manager can instantly know the status of the involvement of a particular resource within the project. The reader is referred to “projects” tab on the right corner. The entire details of the project like project personnel, issue status (open, resolved, closed, in progress), priority, milestones, categories are displayed. Information about project personnel, and the issues assigned/reported by him/her are displayed under the “issue filter” category. Issue summary statistics can be obtained instantly by clicking on the “My Issues” tab provided in the tool, which displays the entire information about the issues that are assigned, reported, in progress, closed and resolved by that particular user. New issues are reported as shown under the new issue tab. The steps involved in the creation of a new project in a common user interface tool follow in the screenshots. The steps involved are quicker compared to Multiple User Interface tool. Similarly a new tab known as “reports” would provide the project manager with day by day status reports (filtered by issues, by status, by version etc.). This is presented the screenshots. Options such as “projects” and “reports” are not instantly available in multiple user interface tool.
A Project Manager needs access to the overall history of ongoing and resolved issues of any project (log documentation about project code). This is essential information required for the project manager to close the project. In this thesis, options (tabs) such as “roadmap” and “reports” are provided which allow the project manager to instantly access required documentation. Roadmap shows the ongoing and resolved issues in context of the percentage of the project completion. In a multiple user interface tool, the project manager is provided with an “administrator” tab which provides the above information after a sequence of required selections like project name, issue status, version, start date; etc. while the new tool developed in this thesis provides all the required information in an organized format under the reports tab. Defect Management and Project Management scenarios are represented in the above example.

A couple of screenshots shown are considered special cases. For example, one represents a screenshot that spans all the above noted four features (task scheduling, log trouble reports into a database, keeping track of project personnel and their assignments, and log documentation about project code). Before gaining access to the login screen, certain errors are encountered within the multiple user interface environment. The errors are discussed in detail under the screenshot. As a general example, we now provide a situation where the simultaneous access to the issue is restricted from multiple users in a multiple user interface tool. In a software organization company, a developer would typically work on resolving the issue reported by the tester. As the developer is continuing to work on the issue, a tester would not be able to simultaneously access the information to the status of the issue in a multiple user interface tool login screen. The tester would be essentially locked out wasting time resources until the developer logs out. While in a common user interface tool, simultaneous access is allowed between the tester and developer prompting the tester to be able to provide necessary information as requested by the developer to resolve the particular issue instantaneously. This additional feature is a welcome relief in the perspective of the time over-head.

The design layout of the tool does not necessarily address any of the four features but is included as an additional benefit to the Project Manager. Current multiple user interface tools do not provide this feature.
In order to test the hypotheses, the following 4 scenarios were established for this thesis – defect capture, defect management, project status management, and security. For the detailed descriptions of each of the 4 scenarios, the reader is referred to Chapter 3. The general work flow for the 4 scenarios is represented in the following figures. As noted the pertinent scenario is identified under each screenshot where applicable.

The work flow process in the Project Status Management scenario is represented in Figure 5.1. The work flow process in the Security scenario is represented in Figure 5.2. The work flow process in the Defect Capture and Defect Management scenarios is represented in Figure 5.3.

Now the results from the implementation of the common user interface tool created in this thesis are presented in the following pages. With each screenshot the applicable feature and the pertinent scenario, the validated hypothesis are noted.

Figure 5.4 represents a simple web based GUI with a common user interface that allows simultaneous access by any team member including PM, and the client. In a multiple user interface environment, while a quality assurance analyst is attempting access to the login screen in-order to work with the issues the following errors are encountered – the tool prompts the user to manually install an “ActiveX control--npmozax.dll” and placing it in the browser plugin directory, downloading the client components manually, and present other access violation errors that can be time consuming to resolve. In the common user interface tool (this thesis), the design incorporated ASP.Net with managed Windows Forms Control feature that supports the “ActiveX control--npmozax.dll” and does not require the user to install or download any component manually.

Cross browser compatibility is also achieved (Firefox). This feature is not available in Multiple User Interface tool for example like “quality center”. Hypothesis 3 has been verified and a relationship between common user interface and cross browser compatibility is established.

Figure 5.5 shows the list of entire projects (example - control errors, database issues, database utilities, new project, and software errors) in the company and represents the project status management scenario. The find tab at the top right corner provides an efficient search mechanism of a particular issue. This information is used by the PM (Administrator) to make the required decisions to manage the project. The PM is provided access to the entire details
Figure 5.1. Project status management scenario workflow.
of a project like assigned project personnel etc., thereby helping the PM in keeping track of
the different project personnel’s work load. The manager assigned to the task is also
identified. In a multiple user interface tool instant access to the list of all projects and issues
is not achieved. While in the above screenshot it is evident that in the new tool with a
common user interface, instant access to all projects, issues, and efficient search capability is
available by clicking on the listed tabs (all, in progress, created recently, updated recently,
assigned to a resource and find). In Figure 5.5, hypothesis 1 and hypothesis 3 are verified.

Cross browser compatibility is also achieved. In order to verify hypothesis 3, the
ability to achieve cross compatibility, the features shown here have been implemented and
tested in both IE and Mozilla Firefox.

Figure 5.6 represents the project status management scenario. This is visible only to
project manager. The PM can delete and manage users, and projects as shown in Figure 5.6.
A project manager is responsible to manage exception details (errors) encountered in any
project within the software organization. As shown, a new option (tab) known as “application
log viewer” has been provided which provides instant access to the exception log and helps
the PM in managing the resolution of the errors. In comparison, a multiple user interface tool
will not provide instant access to the exceptions log that is encountered within a database.

With Figure 5.6 screenshot, hypothesis 1, 2 and 3 are verified. PM has instant access
to the required information thereby reducing time spent (as with the case in a multiple user
Figure 5.3. Defect capture and defect management workflow.
Figure 5.4. Login screen.
Figure 5.5. Administrator main page.
Figure 5.6. Administration page.
interface tool). In order to verify hypothesis 3, the ability to achieve cross compatibility, the features shown here have been implemented and tested in both IE and Mozilla Firefox.

In Figure 5.7 we present the ability of the PM to change the design layout utilizing the basic, authentication settings shown. In a Multiple User Interface tool, the design layout is fixed. The PM is unable to change the layout per his or her requirements (wishes). This represents the project status management scenario. This is visible only to project manager. Figure 5.7 shows how the PM can manage the design layout of a webpage while also enforcing security.

Security scenario is also addressed in Figure 5.7. Compared to a Multiple Use Interface tool, we have provided efficient security mechanism by provision of the following options. As shown in the screenshot, the options like WindowsSAM, ActiveDirectory, and none are provided which enable secure authentication access to the login screen.

Hypothesis 3 is verified in Figure 5.7 screenshot. In order to verify hypothesis 3, the ability to achieve cross compatibility, the features shown here have been implemented and tested in both IE and Mozilla Firefox.

Figure 5.8 shows an error generated within the database when the tool is used for a software project. This error was used to test the time overhead hypothesis. When the project manager clicks the “exception” shown in the screenshot, the error code is displayed instantly. In a Multiple User Interface tool, the exception log is instantly not available. This data accessed by the project manager in this screenshot is especially applicable in the context of “log trouble reports into a database and claimed fixes” discussed already in the above sections of this chapter.

Figure 5.9 represents Project Status Management Scenario. This displays entire information of a project so that the PM has access to the detailed description about the project including issue status, priority, type, category, milestones, resources allocated to work on the issue. Thereby time overhead is reduced. In a Multiple User Interface tool, simultaneous access to the entire information shown above is not available, instead the user to follow multiple steps before access is provided to the desired data.

Also, cross browser compatibility is achieved. Hypothesis 1 and 3 are verified. In order to verify hypothesis 3, the ability to achieve cross compatibility, the features shown here have been implemented and tested in both IE and Mozilla Firefox.
Figure 5.7. Application configuration page.
Figure 5.8. Application log viewer.
Figure 5.9. Project – software issues.
Figure 5.10 represents security and project management scenarios and is only visible to project manager. This feature shows the ability of a PM in either deleting or inactivating a user instantly from being involved in a project. As already noted in Chapter 1, task scheduling by a Project Manager involves creating a new project, assigning appropriate resources, defining roles and responsibilities (delete and edit users and managing their roles and responsibilities), and estimating a project completion date. A Project Manager assigning appropriate resources utilizing a common user interface tool has instant access to the resources while the instant access feature is unavailable to the Project Manager in a multiple user interface tool. Thus, using a common user interface tool leads to reduction in time over-head compared to a multiple user interface tool.

With Figure 5.10 screenshot, hypothesis 2 and 3 are verified. In order to verify hypothesis 3, the ability to achieve cross compatibility, the features shown here have been implemented and tested in both IE and Mozilla Firefox.

Figure 5.11 represents security and project management scenarios and is only visible to the project manager (PM). This shows how a PM can edit the credentials of a team member (user). Especially this screenshot shows the ability of a PM to “authorize” a specific resource from being involved in a project. This verifies hypothesis 1 and 2 by reducing time overhead and providing instant resource allocation.

Hypothesis 3 is also verified. Also, as shown in Figure 5.11 screenshot cross browser compatibility is achieved.

Figure 5.12 represents three scenarios – defect capture, defect management, and project status management. A Project Manager needs to keep track of the different project personnel work loads in-order to assign new projects as and when the personnel are completed with their assignments (keeping track of project personnel and their assignments). Figure 5.12 shows the tool’s ability to provide an efficient search capability for issues or defects. By using “Issue filter” option, efficient search of an issue or a defect (example – user, reporter, type, active status, resolution, priority, version, milestone, category, keyword search, and issue ID) is achieved thereby reducing the time overhead. Hypothesis 1 and 3 are verified.

In order to verify hypothesis 3, the ability to achieve cross compatibility, the features shown here have been implemented and tested in both IE and Mozilla Firefox.
Delete User - admin

It is highly recommended that you unauthorized the user account instead of deleting to maintain the integrity of the APSA database.

Unauthorized the account will keep the user information intact but will not allow the user to log into the application. You can re-authorize the account at anytime if necessary from the user details page.

Unauthorized this Account  Delete this User

Figure 5.10. Delete user.
**Figure 5.11. Edit user.**
**Figure 5.12. Issue list.**

<table>
<thead>
<tr>
<th>Id</th>
<th>Type</th>
<th>Priority</th>
<th>Summary</th>
<th>Version</th>
<th>Milestone</th>
<th>Category</th>
<th>Status</th>
<th>Assigned To</th>
<th>Reporter</th>
<th>Due Date</th>
<th>Last Update</th>
<th>Created Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>102-40</td>
<td>Bug</td>
<td>High</td>
<td>Issue with sending a mail with video attachment, while in call</td>
<td>2.0</td>
<td>2.0</td>
<td>Software Errors</td>
<td>Open</td>
<td>petrick</td>
<td>admin</td>
<td>2/12/2012</td>
<td>2/12/2012</td>
<td></td>
</tr>
<tr>
<td>102-46</td>
<td>Bug</td>
<td>High</td>
<td>Reset occurred when Navigation running in the background on DUT</td>
<td>2.0</td>
<td>2.0</td>
<td>Software Errors</td>
<td>Open</td>
<td>russell</td>
<td>admin</td>
<td>2/13/2012</td>
<td>2/13/2012</td>
<td></td>
</tr>
<tr>
<td>102-51</td>
<td>Bug</td>
<td>High</td>
<td>DUT Lock up while in call log</td>
<td>2.0</td>
<td>2.0</td>
<td>Software Errors</td>
<td>Open</td>
<td>petrick</td>
<td>admin</td>
<td>2/12/2012</td>
<td>2/12/2012</td>
<td></td>
</tr>
<tr>
<td>102-48</td>
<td>Bug</td>
<td>High</td>
<td>Soft Reset occurred while in Skype audio call on DUT</td>
<td>2.0</td>
<td>2.0</td>
<td>Software Errors</td>
<td>Open</td>
<td>petrick</td>
<td>admin</td>
<td>2/12/2012</td>
<td>2/12/2012</td>
<td></td>
</tr>
<tr>
<td>102-73</td>
<td>Bug</td>
<td>High</td>
<td>Random error while in video player</td>
<td>2.0</td>
<td>2.0</td>
<td>Software Errors</td>
<td>Open</td>
<td>petrick</td>
<td>admin</td>
<td>2/12/2012</td>
<td>2/12/2012</td>
<td></td>
</tr>
<tr>
<td>102-31</td>
<td>Bug</td>
<td>High</td>
<td>Introduction of WiFi Calling feature</td>
<td>2.0</td>
<td>2.0</td>
<td>Software Errors</td>
<td>Open</td>
<td>petrick</td>
<td>admin</td>
<td>2/12/2012</td>
<td>2/12/2012</td>
<td></td>
</tr>
<tr>
<td>102-49</td>
<td>Bug</td>
<td>High</td>
<td>Random error while watching youtube video</td>
<td>2.0</td>
<td>2.0</td>
<td>Software Errors</td>
<td>Open</td>
<td>petrick</td>
<td>admin</td>
<td>2/12/2012</td>
<td>2/12/2012</td>
<td></td>
</tr>
<tr>
<td>102-52</td>
<td>Bug</td>
<td>High</td>
<td>Issue Messages are not being received</td>
<td>2.0</td>
<td>2.0</td>
<td>Software Errors</td>
<td>Open</td>
<td>petrick</td>
<td>admin</td>
<td>2/13/2012</td>
<td>2/13/2012</td>
<td></td>
</tr>
<tr>
<td>102-47</td>
<td>Bug</td>
<td>High</td>
<td>Random error while in Calendar App on DUT</td>
<td>2.0</td>
<td>2.0</td>
<td>Software Errors</td>
<td>Open</td>
<td>petrick</td>
<td>admin</td>
<td>2/12/2012</td>
<td>2/12/2012</td>
<td></td>
</tr>
<tr>
<td>102-36</td>
<td>Bug</td>
<td>High</td>
<td>Random error occurred on Device Under Test</td>
<td>2.0</td>
<td>2.0</td>
<td>Software Errors</td>
<td>Open</td>
<td>petrick</td>
<td>admin</td>
<td>3/12/2012</td>
<td>2/12/2012</td>
<td></td>
</tr>
</tbody>
</table>
Figure 5.13 covers defect management scenario. A Project Manager needs to keep track of the different project personnel work loads, issues reported by the personnel in-order to assign new projects as and when the personnel are completed with their assignments (keeping track of project personnel and their assignments). This shows the tools ability to provide instant access to issue summary statistics thereby reducing time overhead. Also, cross browser compatibility is achieved. For example, Issue summary statistics can be obtained instantly by clicking on the My Issues tab provided in the tool, which displays the entire information about the issues that are assigned, reported, in progress, closed and resolved by that particular user. Hypothesis 1 and 3 are verified. Cross-browser compatibility is achieved as shown in Figure 5.13 screenshot.

Task scheduling by a Project Manager involves creating a new project, assigning appropriate resources, defining roles and responsibilities (modify credentials of a team member), and estimating a project completion date are shown in Figure 5.14. A Project Manager assigning appropriate resources utilizing a common user interface tool has instant access to the resources while the instant access feature is unavailable to the Project Manager in a multiple user interface tool. Thus, using a common user interface tool leads to reduction in time over-head compared to a multiple user interface tool. Figure 5.14 shows the ability of the PM to modify the credentials of a team member. Figure 5.14 also represents security and project management scenarios and is only visible to a project manager. Hypothesis 3 is verified by establishing cross browser compatibility as shown in Figure 5.14 screenshot.

Figure 5.15 shows the defect capture scenario. As noted, a Project Manager needs to keep track of the different project personnel work loads in-order to assign new projects as and when the personnel are completed with their assignments (keeping track of project personnel and their assignments). The “new issue” tab allows the team members and the PM to report a new issue by providing the issue details such as summary, version, type, category, priority, resolution hours, description, attachments, and assignment to an appropriate resource; In a multiple user interface tool, assignment to an appropriate resource is performed automatically by the tool while in a common user interface this function is available to the PM or user. This reduces time overhead in situations where the tool automatically assigns to the inappropriate resource. Hypothesis 1 and 3 are verified. Cross browser compatibility is achieved as shown in Figure 5.15 screenshot.
**Figure 5.13. Issue summary.**
Figure 5.14. Manage roles for user.
Figure 5.15. New issue.
Figures 5.16 to 5.22 cover the project status management scenario. This is available to the PM only for access purposes. Hypothesis 1, 2 and 3 are verified in the below screenshots. Before a new project is created the Project Manager will typically follow certain steps. Figures 5.16 through 5.22 show the required steps in a sequential manner that is involved in the creation of a new project. Figure 5.16 represents the first step in the creation of a new project. Figure 5.17 shows the second step in which the PM fills out project name, details, project code and access type. Figure 5.18 requires the PM to add or select the project category. Figure 5.19 enables the PM to fill the project version. Figure 5.20 allows the PM to define the user roles (project administrator, tester, developer, quality analyst, and client) in a project. Figure 5.21 allocates the members to the already defined user roles. Figure 5.22 confirms the completion of a new project creation. All the above steps are comparatively quicker and easier in a Common User Interface tool compared to a Multiple User Interface tool. Options like instant resource allocation, assigning roles to the resources, project unique code, and access type are available to the PM while creating a new project. This reduces time overhead and provides instant allocation.

In order to verify hypothesis 3, the ability to achieve cross compatibility, the features shown here have been implemented and tested in both IE and Mozilla Firefox.

Figures 5.23 and 5.24 cover the project management scenario. A Project Manager needs to keep track of the different project personnel work loads in-order to assign new projects as and when the personnel are completed with their assignments (keeping track of project personnel and their assignments). This shows the ability of the tool to provide the PM with an instant access to day by day status report on a project. Hence, time overhead is reduced by providing Reports - Categories tab in the tool, which allows the project managers to instantly access the day by day status reports (filtered by issues, by status, by version etc.). Hypothesis 1 and 3 are verified. In order to verify hypothesis 3, the ability to achieve cross compatibility, the features shown here have been implemented and tested in both IE and Mozilla Firefox.

Figure 5.25 covers the project management status scenario. A Project Manager needs access to the overall history of ongoing and resolved issues of any project (log documentation about project code). This is essential information required for the project manager to close the project. Roadmap shows the ongoing and resolved issues in context of
Figure 5.16. New project –step I.
Figure 5.17. New project –step II.
Figure 5.18. New project – step III.
Figure 5.19. New project –step IV.
Manage Security Roles

Each project can have its own roles to group the users and permissions. By default when a new project is created, several roles are predefined for you. If you need to create a custom role, click the ‘add new role’ button, then assign permissions to the role.

Add New Role

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Auto Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Administrator</td>
<td>Project Administrators</td>
<td></td>
</tr>
<tr>
<td>Read Only</td>
<td>Read Only</td>
<td></td>
</tr>
<tr>
<td>Reporter</td>
<td>Reporter</td>
<td></td>
</tr>
<tr>
<td>Developer</td>
<td>Developer</td>
<td></td>
</tr>
<tr>
<td>Quality Assurance</td>
<td>Quality Assurance</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.20. New project – step V.
Figure 5.21. New project – step VI.
Figure 5.22. New project –step VII.
### Figure 5.23. Reports.

**Software Issues**  
**Date: 2/10/2012**

**Open**

<table>
<thead>
<tr>
<th>ID</th>
<th>Type</th>
<th>Summary</th>
<th>Priority</th>
<th>Assigned To</th>
<th>Reporter</th>
<th>Last Update</th>
</tr>
</thead>
<tbody>
<tr>
<td>J02-40</td>
<td>Any</td>
<td>Issue with sending a mail with video attachment, while in call</td>
<td>Blocker</td>
<td>patrick</td>
<td>admin1</td>
<td>12/02/2012</td>
</tr>
<tr>
<td>J02-46</td>
<td>Improvement</td>
<td>Reset occurred when Navigation running in the background on DUT</td>
<td>Blocker</td>
<td>russell</td>
<td>admin1</td>
<td>13/02/2012</td>
</tr>
<tr>
<td>J02-51</td>
<td>Any</td>
<td>DUT Lock up while in call log</td>
<td>Blocker</td>
<td>patrick</td>
<td>admin1</td>
<td>13/02/2012</td>
</tr>
<tr>
<td>J02-48</td>
<td>Improvement</td>
<td>Soft Reset occurred while in Skype audio call on DUT</td>
<td>Blocker</td>
<td>stephen</td>
<td>admin1</td>
<td>13/02/2012</td>
</tr>
<tr>
<td>J02-73</td>
<td>Any</td>
<td>Reset occur while in idio player</td>
<td>Blocker</td>
<td>russell</td>
<td>victor</td>
<td>13/02/2012</td>
</tr>
<tr>
<td>J02-39</td>
<td>New Feature</td>
<td>Introducing the Wifi Calling feature</td>
<td>Critical</td>
<td>stephen</td>
<td>admin1</td>
<td>12/02/2012</td>
</tr>
<tr>
<td>J02-49</td>
<td>Improvement</td>
<td>Reset while watching youtube video</td>
<td>Critical</td>
<td>stephen</td>
<td>admin1</td>
<td>13/02/2012</td>
</tr>
<tr>
<td>J02-52</td>
<td>Bug</td>
<td>Enclose Messages are not being received</td>
<td>Critical</td>
<td>peter</td>
<td>admin1</td>
<td>13/02/2012</td>
</tr>
</tbody>
</table>
Figure 5.24. Issues by status.
Figure 5.25. Road map.
the percentage of the project completion. Figure 5.25 shows the ability of the tool to provide the PM with the project progress information (status on the right shown as a percentage). In a multiple user interface tool, the project manager is provided with an “administrator” tab which provides the above information after a sequence of required selections like project name, issue status, version, start date; etc. while the new tool developed in this thesis provides all the required information in an organized format under the reports tab. Hence, time overhead is reduced by providing a Roadmap tab in the tool to project managers which allows them to instantly have project progress information. Hypothesis 1 and 3 are verified. In order to verify hypothesis 3, the ability to achieve cross compatibility, the features shown here have been implemented and tested in both IE and Mozilla Firefox.
CHAPTER 6

RESEARCH CONCLUSIONS AND APPLICATIONS TO PRACTICE

6.1 RESEARCH CONCLUSIONS

The results from the implementation of the new project management tool have led to the following conclusions. We hypothesized that a direct relationship exists between a common user interface and time overhead reduction. The tool implemented under all four scenarios, established a direct relationship between common user interface and time overhead reduction. With the new tool a reduction in time was observed in performing similar tasks/troubleshooting issues compared to a multiple user interface tool. Under hypothesis 2, we hypothesized that a direct relationship exists between a common user interface and instant resource allocation. The tool implemented for the project management status scenario established a direct relationship between common user interface and instant resource allocation. With the new tool the PM was able to instantly review information, manage information and appropriately make decisions. Under hypothesis 3, we hypothesized that a direct relationship exists between a common user interface and browser cross compatibility. We also concluded that the tool implemented under all four scenarios, established a direct relationship between common user interface and cross browser compatibility (Mozilla Firefox and Internet Explorer). The new common user interface project management tool (this thesis) is concluded to be superior to other multi user interface project management tools (although we were able to locate one only, since there might be other proprietary tools used by companies that are confidential) available in the market. The new tool is shown to have more features and can be implemented as a low cost alternative.

The following section compares the existing tools in the market with the new tool created in this thesis. Quality Center is a multiple user interface tool that provides complete end to-end quality management process, is very expensive ($60,000) and not a very user friendly tool for testers [21]. This is not affordable for 99% of the companies. Microsoft
Project Sever and Active Collab are the two other common user interface tools that are used in the market. Microsoft Project Server provides comprehensive solutions with most of the features along with timesheet and project tracking tool at $7,000. Active Collab provides project management and collaboration tool, time tracking, ticket management and milestones at $199 per year. The new project management tool created in this thesis provides additional features like project management, time tracking, issue tracking, instant resource allocation, exception capture, efficient search mechanism (people finder), and provides an efficient search filter for issues/defects.

6.2 APPLICATIONS TO PRACTICE

The rationale behind this work was to create a simple web based common user interface software project management tool that reduces time overhead, provides instant access to resources, and improves security while also exhibiting cross compatibility with all the browsers. This tool is especially targeted for software organizations as a low cost alternative to other existing common user interface tools available in the market.
CHAPTER 7

FUTURE WORK

The new software project management tool developed in this thesis project incorporates the following features listed under Section 1.2. The features include: tasks scheduling, log trouble reports into a database, versioning (manual), keeping track of project personnel and their assignments, and log documentation about project code. Unit testing is a method by which individual units of source code are tested to determine if they are fit for use. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended. Its implementation varies from being very manual (pencil and paper) to being automated [22]. Pertaining to the thesis the developed tool supports manual unit testing.

Suggestions for future work include incorporating additional features such as unit testing coordinated with versioning, refactoring coordinated with versioning, unit testing automation and deployment of this application to support mobile environment.
BIBLIOGRAPHY


