CISC 615 Software Testing and Maintenance

Mid Term Project

Jingda Guo

Feng Mi

Weiguo Lin

Shengling Qiu

Weida Zhu

Contents

[1 Testing Strategy Document 1](#_Toc433320543)

[1.1 Introduction to People Resources Project 1](#_Toc433320544)

[1.2 Purpose of This Document 1](#_Toc433320545)

[1.2.1 Approaches 1](#_Toc433320546)

[1.2.2 Timeline 5](#_Toc433320547)

[2 Implementation Test Plan 5](#_Toc433320548)

[3 Resource Requirements for Implementation 5](#_Toc433320549)

[3.1 Human Resources 5](#_Toc433320550)

[3.2 Hardware Resources 6](#_Toc433320551)

[3.3 Software Resources 6](#_Toc433320552)

[3.3.1 Visual Studio 2015 Internal Tools 6](#_Toc433320553)

[3.3.2 DbUnit 7](#_Toc433320554)

[3.3.3 QuickTest Professional (QTP) 7](#_Toc433320555)

[3.3.4 DataFactory 7](#_Toc433320556)

[3.3.5 Loadrunner 7](#_Toc433320557)

[3.4 Budget 8](#_Toc433320558)

[4 Resource Requirements for Ongoing Maintenance Testing 8](#_Toc433320559)

# 1 Testing Strategy Document

## 1.1 Introduction to People Resources Project

People Resources Project is an HR application providing a platform of services for small organizations across the United States for all facets of human resource processing including recruitment, employee data, job functions, employee benefits, payroll, and managing the employee life cycle from pre-hire to termination. The application will be web-based and be developed in Visual Studio 2015 with MS SQL Server as the data repository. It will also follow the scrum methodology, which means that a scrums of 5 scrums will be used to integrate the 5 different scrum teams.

## 1.2 Purpose of This Document

The purpose of this test strategy document is to clarify the overall targets, approach, tools and timing of test activities. The overall target is to rewrite a new system and test the functionality of the system with the scrum methodology.

### 1.2.1 Approaches

*Unit Testing*

Dispatch one member from each testing team (5 persons in total) and embed them into the development team of the five functions to do the test-driven development testing together with the developer during the development process. After new code is written, testing engineers or technical architects will utilize unit testing to ensure the 1 million lines of code are developed correctly. The unit testing will be automatically executed with the internal testing tool in Visual Studio 2015.

Use hybrid automation scripts which maximize the keyword-driven, data-driven and structure scripting in order to make modification easier and keep the script intact to the maximum extent while updating.

Pay more attention and spend more time on concise, modular and high code-reusability scripting and there would be several times compensation in running and updating them.

*Component Integration Testing*

Conduct quick component testing to ensure each main component function properly before the formal component integration testing begins, i.e. recruitment, employee data, job functions, employee benefits and payroll.

Use bottom-up testing strategy in component integration testing to avoid ignoring even small defects of the interaction of each component.

Component integration testing of each function should be tested automatically using data-driven type automation to focus on the handling of data passing between each component.

*System Integration Testing*

Make sure the high quality and the completion of unit testing before the integration process begins and make big-bang testing first before the formal system-integration testing being executed.

System-integration testing should be tested manually by using top-down testing strategy.

*System Testing*

Several types of system testing need to be conducted to ensure that the whole system works when integrated.

Smoke testing will be done before any other system testing. It is the least expensive way to see whether all the crucial functions work well, but not bothering with finer details as it only checks the most important functions of the application.

Migration testing is done to ensure that the application can be successfully moved from older system infrastructures to current system infrastructures.

Hardware testing is utilized to test the compatibility of the application with hardware (different types of PCs, mobile phones and tablets).

Functional testing, which involves trying to think of any possible missing functions, is conducted to examine the five main functions of the application and to avoid any omissions of additional functions.

Load testing and stress testing are used to simulate both real-life normal loads and extremely high flow rate to see whether the five functions of the application can work properly.

Recovery testing is done to demonstrate that a solution is reliable, trustworthy and can successfully recoup from possible crashes.

Data back-up is done to ensure that data would not be lost when unexpected natural disasters or human mistakes happen.

*Database Testing*

It needs to be tested whether our database meet the functional or the stress level by use different automation tools. When we check the function level of the database we use DbUnit, Quick Test Professional, DataFactory. We use this tools to check input and output of our database. To test the stress level of the database, Loadrunner can simulate the situation that large amounts of people use this database to check whether the database run normally.

*Alpha Testing*

A final process before the software release to public. Bugs, crashes, missing docs and features are expected to be find in Alpha testing. Testers are going to focus on the stability and adaptability of the software in different intended environment and automation testing tools will be used to improve the software quality.

*Users Acceptance Testing*

A new QA team is assigned to imitate customers’ behavior in the software. Testing will focus on the main functions of the requirement and the humanization and suitable for most customers. The aesthetic measure will also be involved in this process.

*Beta Testing*

Releasing the software to selected public, Check whether there are any underlying bugs in the software, the integrity of the information showed to customers or the stability of the connection between users and servers. All the previous requirements and specifications will be tested in this process.

## 1.2.2 Timeline

|  |  |
| --- | --- |
| Testing Steps | Time Planned |
| Unit Testing | 20-22 weeks |
| Component Testing | 20-22 weeks |
| Integration Testing | 4-5 weeks |
| System Testing | 8 weeks |
| Integration Testing | 8 weeks |
| User Acceptance Testing | 4 weeks |
| Alpha Testing | 2 weeks |
| Beta Testing | 4 weeks |

# 2 Implementation Test Plan



# 3 Resource Requirements for Implementation

## 3.1 Human Resources

The test will be a one year process. 5 scrum teams of eight people and a small QA team of three people (about 6 weeks) are required in testing.

Salaries for the engineers (as an average salary of software engineer in 2014):

Software testers: 5\*8\*93,300=$3,732,000

QA team: 3\*10,367=$31,100

Total human resource cost:

$3,732,000+$31,100=$3,763,100

## 3.2 Hardware Resources

When the testing is initialized, hardware is our basic need.

Our hardware resources requirements:

* Testing environment that meets the needs
* PCs (with necessary Accessories) running on Mac and Windows to test whether the web application which is built fits different working systems
* Smart phones and tablets, e.g. IOS, Android, Windows Phone

Testing should include these mobile equipment that users may use to access the web application. Make sure different types of browsers can get the same accurate information, e.g. Internet Explorer, Safari, Firefox and Chrome.

Once testing environments are satisfied, move on to make sure the data store server (the place where the data is stored) and the web server (the place where the application runs on) are well intact and function well.

A firm connection between server and client should be ensured and tested.

## 3.3 Software Resources

### 3.3.1 Visual Studio 2015 Internal Tools

Performance testing for the application is made easier with Visual Studio tools. Scale tests to hundreds of thousands of concurrent users and generate load from multiple regions worldwide. Our team can test changes continuously in a fast, scalable, and efficient manner based on the technologies and frameworks. Besides, it allows us to get end-to-end traceability across all artifacts, run tests and log defects from browsers or use the Microsoft Test Manager (MTM) client so it can guarantee assessing quality throughout testing lifecycle.

### 3.3.2 DbUnit

DbUnit can work with very large datasets when used in streaming mode. It can also help the test team to verify that database data match an expected set of values.

### 3.3.3 QuickTest Professional (QTP)

QTP provides functional and regression tests automation for software applications and environments. Unified Functional Testing can be used for enterprise quality assurance. Unified Functional Testing supports keyword and scripting interfaces and features a graphical user interface.

### 3.3.4 DataFactory

DataFactory can Compose and orchestrate data services at scale.

### 3.3.5 Loadrunner

Loadrunner is used to test applications, measuring system behavior and performance under load. LoadRunner can simulate thousands of users concurrently using application software, recording and later analyzing the performance of key components of the application.

## 3.4 Budget

Besides the human resource cost, the cost of other resources should also be included in the budget. The software required in the testing are charged free.

|  |  |  |  |
| --- | --- | --- | --- |
| Items | Unit Cost | Units | Total Cost |
| **PCs** |  |  |  |
| Mac | $1,499 | 2 | $2,998 |
| Windows | $1,000 | 2 | $2,000 |
| **Mobile Phones** |  |  |  |
| IOS phone | $849 | 2 | $1,698 |
| Android phone | $699 | 2 | $1,398 |
| Windows phone | $539 | 2 | $1,078 |
| **Tablets** |  |  |  |
| IOS tablet | $599 | 2 | $1,198 |
| Android tablet | $429 | 2 | $858 |
| Windows tablet | $469 | 2 | $938 |
| **Human Resources** |  |  | $3,763,100 |
| **Total Budget** |  |  | **$3,775,266** |

# 4 Resource Requirements for Ongoing Maintenance Testing

When the application is finally put into market, maintenance should take place [immediately](http://dict.youdao.com/w/immediately/). Modifications and updates should be made to get rid of bugs and to adapt to new requirements. It can be performed at either the system level, or the component level. Maintenance testing uses system performance requirements as the basis for identifying the appropriate components for further inspection or repair. What is more, in order to route changes from development into production, keeping touch with the customers and see what kind of defects they detect is a high-blow. It should obey three steps that keep making a survey and collect information from the website of the application to make sure the defects which people meet. An estimation, whether the defects could be handled and whether the defects are true, will then be made. Finally, changes will be made and the system patch will be updated.

A testing team should keep monitoring the whole system and finding bugs, while another team handles the modification. Before the modifications are provided for users to update, a testing team should make sure the modifications are bug free.

With the help of Planned Maintenance and Condition Based Maintenance, we can employ preventive maintenance method to provide systematic inspection, [detection](https://en.wikipedia.org/wiki/Fault_detection_and_isolation), and correction of incipient failures before they occur or before they develop into major defects.

Resources needed in this process:

(1) Three teams and at least 5 people in each team.

(2) Funding to do the survey and the information will be collected manually, which depends on how the company operates at that time.