|  |
| --- |
| Test Plan |
| Human Resource Management |
| Descript about testing, test-cases, test process… |

6/6/2012

Table of Contents

[Introduction 4](#_Toc326766513)

[1.1 Purpose 4](#_Toc326766514)

[1.2 Scope 4](#_Toc326766515)

[1.3 Project Overview 4](#_Toc326766516)

[1.4 Test Phases 5](#_Toc326766517)

[1.4.1 Integration testing 5](#_Toc326766518)

[1.4.2 System testing: 5](#_Toc326766519)

[1.4.3 Acceptance testing: 6](#_Toc326766520)

[1.5 Test Result 6](#_Toc326766521)

[1.5.1 Integration testing: 6](#_Toc326766522)

[1.5.2 System testing: 6](#_Toc326766523)

[1.5.3 Acceptance testing: 6](#_Toc326766524)

[Test Schedule 7](#_Toc326766525)

[Human Resource 8](#_Toc326766526)

[3.1 Team Information 8](#_Toc326766527)

[3.2 Training Schedule 8](#_Toc326766528)

[3.3 Roles & Responsibilities 8](#_Toc326766529)

[Test Environment 10](#_Toc326766530)

[4.1 Hardware 10](#_Toc326766531)

[4.2 Software 10](#_Toc326766532)

[4.3 Test tools 10](#_Toc326766533)

[Using Internet Explorer browser to test system testing 10](#_Toc326766534)

[4.4 Test Databases 10](#_Toc326766535)

[All test results must be saved in the Configuration Management System. 10](#_Toc326766536)

[Document Deliverables 11](#_Toc326766537)

[Entrance & Exit Criteria 11](#_Toc326766538)

[6.1 Entrance Criteria 11](#_Toc326766539)

[6.1.1 Integration testing 11](#_Toc326766540)

[6.1.2 System testing 12](#_Toc326766541)

[6.1.3 User acceptance testing (optional) 12](#_Toc326766542)

[6.2 Exit Criteria 12](#_Toc326766543)

[6.2.1 Integration testing 12](#_Toc326766544)

[6.2.2 System testing 13](#_Toc326766545)

[6.2.3 User acceptance test (optional) 13](#_Toc326766546)

[Defect Life Cycle 13](#_Toc326766547)

[7.1 Defined Defect Life Cycle 13](#_Toc326766548)

[7.2 Defect Life Cycle Description 14](#_Toc326766549)

[7.3 Guidelines On Deciding The Severity Of Bug 14](#_Toc326766550)

[Appendix Definition 15](#_Toc326766551)

**Revision Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Author** | **Date** | **Reason for changes** | **Version** |
| Tung Nguyen | 16/10/2011 | Add Test Plan | 1.0 |
| Dang Nguyen | 19/10/2011 | Update ENTRANCE & EXIT CRITERIA | 1.0.1 |
|  |  |  | 1.0.2 |

# Introduction

This is the Software Test Plan for the HRM (Human resource Management), which is being produced for our client by Team VLU-HRM.Team VLU-HRM is the development team responsible for both building and testing the code. Team VLU-HRM will deliver the final system to the client on April 17, 2011

## 1.1 Purpose

This document is aimed to give a detailed plan of test strategy, test schedule, resources, environment for performing Unit test, Integration test , System test and Acceptance test of HRM.

This Software Test Plan for the HRM supports the following objectives.

* To detail the activities required to prepare for and conduct the software tests
* To communicate to all responsible parties the tasks which they are to perform
* To define the sources of information used to prepare the plan
* To define the test tools and environment needed to conduct the software tests
* To define the acceptance criteria, as agreed to by the client

## 1.2 Scope

This document describes the plan and procedures that will be used by Team VLU-HRM to verify that the HRM performs as specified in the Software Requirements Specification (SRS). The scope of this document is to cover development and testing of the HRM in 1 module Human Resource Record Management. As the development cycle will be compressed into these 244 days, it is necessary to have a comprehensive and systematic plan for testing the code units as they are developed and integrated into the overall system. This plan will address all testing that will occur.

The test scope includes the following:

* Testing of all functional, application performance, security and use cases requirements listed in the Use Case document.
* Quality requirements and fit metrics HRM
* End-to-end testing and testing of interfaces of all systems that interact with the HRM.

## 1.3 Project Overview

The Human resource management includes the processes required to coordinate the human resources on a project. Such processes include those needed to plan, obtain, orient, assign, and release staff over the life of the project.

HRM is particularly developed for human resource management in university / colleges. The system consists of key modules:

* Personal information management
* Employee labor contract management
* Recruitment & training processing
* Payroll
* Administration panel – Utilities
* Insurance management
* Assessment management
* Income management
* Report management

## 1.4 Test Phases

* Software testing can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed.
* Testing of an application includes:
* Integration testing
* System Testing
* Acceptance testing

### 1.4.1 Integration testing

* All code units will be scheduled for integration testing based on the overall test schedule. Integration testing will focus on testing the interfaces between code units, components, and subsystems (Module). Integration tests will also be run manual. Any defects found during integration testing will be entered into the defect tracking spreadsheet. It is the responsibility of the developer who wrote the code module to find and fix the defect. If it is not known which module produces the defect, the Test Manager will assign the task of finding the defect to a team member. Once the defect has been found and fixed, the integration test must be repeated. Once the integration test has been completed successfully, the test will become part of the regression test suite.
* The person responsible for running the integration test will be either the Test Manager or a team member designated to be responsible for the test. This assignment of responsibility will be included on the overall test schedule. The Test Manager, however, may reassign tests based on the current workload throughout development.
* There are four types of testing in the Integration Test:
* Structure Test
* Functional Test
* Stress Test

### 1.4.2 System testing:

* System testing will be done on the releases prior to delivery to the client and mentors. This system test will include coverage of all functional requirements and quality attribute scenarios covered by the completed code. The Test Manager has responsibility for the system test. Any defects found during system testing will be entered into the defect tracking spreadsheet.
* The Test Manager will assign the task of finding the defect to a team member. Once the defect has been found and fixed, the system test must be repeated. Once the system test has been completed successfully, the software will be packaged for release to the customer.
* There are many types of testing in the system Test, the most popular are:
* Functional Test
* Configuration Test
* Security Test

### 1.4.3 Acceptance testing:

* Acceptance test will test by customer, it is a test conducted to determine if the requirements of a specification or contract are met. It is final step of testing, acceptance tests are generally performed as "black box" tests, and the tester uses specified inputs into the system and verifies that the resulting outputs are correct, without knowledge of the system's internal workings.
* Associated with Acceptance Test phase is usually a group of services and documentation.

## 1.5 Test Result

### 1.5.1 Integration testing:

* System is executed Integration testing and meets requirements of integrating components in SDS.
* Make sure Unit test must be tested fully and completely.
* Integration Test cases and Test reports must be clear and full. These documents will be updated continuously in project lifecycle.
* Bug log report and outstanding bug list.
* Status of bugs and bug lifecycle must be reported regularly.

### 1.5.2 System testing:

* Qualities attributes in SRS document must be tested fully.
* Functionalities that are described in use cases of SRS document must be tested fully and completely.
* System test cases and test reports must be completed fully. These documents will be updated continuously in the project development cycle
* Bug log report and outstanding bugs list
* The status of the bug and bug life cycle must be reported regularly.

### 1.5.3 Acceptance testing:

* Acceptance test cases must be executed fully at the work environment of user.

# Test Schedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Task description** | **Start date** | **Finish date** | **Assigned to** |
| **Test documentation** | | | | |
| 1 | HRM Test Plan | 30/9/2011 | 1/10/2011 | Tung Nguyen |
| **Integration testing (In-house testing)** | | | | |
| 1 | Execute HRM Integration testing |  |  | All Members |
| 2 | Report HRM Integration testing |  |  | All Members |
| **System testing (In-house testing)** | | | | |
| 1 | Execute HRM System testing | 15/2/2012 | 3/3/2012 | All Members |
| 2 | Report HRM System testing | 1/5/2012 | 5/5/2012 | All Members |
| **Acceptance testing (In-house testing)** | | | | |
| 1 | Execute HRM user Acceptance testing | 24/1/2012 | 15/2/2012 | All Members |
| 2 | Report HRM User Acceptance Testing | 5/5/2012 | 15/5/2012 | All Members |

# Human Resource

## 3.1 Team Information

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Name | Role | Team | Email |
|  | Huỳnh Thị Hồng Nhung | Team Leader | HRM team | Email: [nhunghuynhthihong@gmail.com](mailto:nhunghuynhthihong@gmail.com) |
|  | Trần Nguyễn Hoàng Tân | Team Member | HRM team | Email: [hoangtanvlu@gmail.com](mailto:hoangtanvlu@gmail.com) |
|  | Nguyễn Kim Tường | Team Member | HRM team | Email: [kimtuongvlu@gmail.com](mailto:kimtuongvlu@gmail.com) |
|  | Đinh Nguyễn Khôi Nguyên | Team Member | HRM team | Email: [shadow141206@gmail.com](mailto:shadow141206@gmail.com) |
|  | Phan Gia Bá Lộc | Team Member | HRM team | Email: [tuongcuop.ali@gmail.com](mailto:tuongcuop.ali@gmail.com) |
|  | Nguyễn Khắc Quyết | Team Member | HRM team | Email: [nguyenkhacquyet89vl@gmail.com](mailto:nguyenkhacquyet89vl@gmail.com) |
|  | Nguyễn Ngọc Tùng | Team Member | HRM team | Email: [haycogang0207@gmail.com](mailto:haycogang0207@gmail.com) |
|  | Nguyễn Tiến Đặng | Team Member | HRM team | Email: [dangnguyen2409@gmail.com](mailto:dangnguyen2409@gmail.com) |

Testing Group has eight members in implementing test documents phase. In System testing phase, testers must be added

* Project Leader: Nhung Huynh
* Test leader: Tung Nguyen
* Tester: All Members
* Document writer: All Members

## 3.2 Training Schedule

Table 2 Training schedule

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Description** | **Start date** | **End date** | **Team members** | **Instructor** |
| 1 | Research & Training | October 1, 2011 | March 22, 2012 | All Members | Mentors |

## 3.3 Roles & Responsibilities

* Project Leader: Responsible for Project schedules and the overall success of the project
* Test leader: Responsible for guidance, plan and track the progress of the work of the test group, reports directly to the PM.
* Tester: execute testing system following testing documents and discuss with testers about fixing bugs, reports to Test leader directly.

Document writer: responsible to develop testing documents (test plan, test cases, test report)

# Test Environment

## 4.1 Hardware

Table 3 Requirement of Hardware

|  |  |  |
| --- | --- | --- |
| **No** | **Test items** | **Minimum configuration** |
| 1 | PC / Laptop | * Hard disk space: About 2 GB (full installation incl. two language modules). About 1 GB (minimal installation incl. two language modules) * Memory: more than 512 MB free memory with default cache settings * CPU: Equal or more than Pentium 4-2.0 GHz |

## 4.2 Software

Table 4 Requirement of Software

|  |  |  |
| --- | --- | --- |
| **No** | **Test items** | **Software** |
| 1 | Work station | * Windows XP SP2 * Windows Vista * Windows 7 |
| 2 | Development languages, Databases, Application server | * C # * MS SQL Server 2005 * Microsoft Visual Studio 2010 |

## 4.3 Test tools

## Using Internet Explorer browser to test system testing

## 4.4 Test Databases

## All test results must be saved in the Configuration Management System.

# Document Deliverables

Table 5 Testing document deliverables

|  |  |  |
| --- | --- | --- |
| **No** | **Item** | **Description** |
| 1 | <Human Resource Management> Test Plan | This document describes Test Strategy and plan when executing Testing phase of Project. |
| 2 | < Human Resource Management > System Test Specification | This document describes all test cases for System test phase when executing Testing of this project. |
| 3 | < Human Resource Management > Integration Test Specification | This document describes all test cases for Integration test phases when executing Testing of this project. |
| 4 | < Human Resource Management> Acceptance Test Specification | This document describes all test cases for Acceptance test phases when executing Testing of this project. |
| 5 | Test report | This document describes the number of test pass and fails when executing test case. |
| 6 | Defect report | This document describes detail defect as type, location, tester |

# Entrance & Exit Criteria

## 6.1 Entrance Criteria

These criteria should be defined before testing phase began and performance by Project Management, Technical Leader and Test Leader.

### 6.1.1 Integration testing

* Integration test case has been approved.
* All others test had been successfully completed
* Integration test cases must be available
* Employee must be available
* Environment and test tools must be set up.
* Test documents must be reviewed
* Outstanding Bugs List

### 6.1.2 System testing

* System test plan has been approved.
* All others test had been successfully completed
* System test cases must be available
* Employee must be available
* Environment and test tools must be set up.
* Test documents must be reviewed
* Outstanding Bugs List

### 6.1.3 User acceptance testing (optional)

* User Acceptance Test Plan and test cases must be approved and available.
* List of outstanding bugs and issue list
* All others test had been successfully completed
* Employee must be available
* Environment test and test tools must be available
* Test documents must be reviewed

## 6.2 Exit Criteria

### 6.2.1 Integration testing

* 100% integration test must be executed
* Pass rate: 95%
* All bugs are found that must repaired and retest must completed.
* Don’t have bugs with Severity1, 2
* Issues list and outstanding bugs list

### 6.2.2 System testing

* 100% System test cases must be executed.
* Pass rate : 95%
* Don’t have bugs with Severity1, 2
* Issues list and outstanding bugs list

### 6.2.3 User acceptance test (optional)

* 100% User Acceptance test cases must be executed and passed if customer approves that the HRM software meets their requirement.
* If some test cases failed, must have a report of issues that software product need to be changed to fix the problems following customer’s expectation.

# Defect Life Cycle

## 7.1 Defined Defect Life Cycle

In software development process, the defect has a life cycle. The defect should go through the life cycle to be closed. A specific life cycle ensures that the process is standardized. The defect attains different states in the life cycle.

The life cycle of the defect can be shown diagrammatically as follows:



## 7.2 Defect Life Cycle Description

The testers or test group will report the bug to the change control board and Project manager.

Statuses include:

* New
* Accepted
* Invalid Or Duplicate
* WontFix
* Started
* Fixed
* Verified
* Done Or WontFix

## 7.3 Guidelines On Deciding The Severity Of Bug

Indicate the impact each defect has on testing efforts or users and administrators of the application under test. This information is used by developers and management as the basis for assigning priority of work on defects.

A sample guideline for assignment of Priority Levels during the product test phase includes:

* Critical
* High
* Medium
* Low

# Appendix Definition

**Table 6: Describing test case priority**

|  |  |
| --- | --- |
| **Priority** | **Description** |
| 1 | Test case is very important and cannot fail. Bug must be fixed if test case fails. |
| 2 | Test case is not important and can be executed or not base on test schedule. |
| 3 | Test case is not important and do not need to test. However, when executing this test case, it will help to improve quality of software. |

**Table 7: Bug Description**

|  |  |
| --- | --- |
| **Status** | **Description** |
| **New** | When the defect is posted for the first time, Tester approves that the bug is new and change the status of defect as “New”. Tester will collect the number of new defect every week.  After the defect is report, we have a special team for analysis the impact and evaluate the benefit of benefit or change if we fix this defect. |
| **Accepted** | If the defect is real and impact on the project, the defect status will be change to “Accepted”  Once the defect is change to “Approved”, lead of testers assign the defect to corresponding developer or developer team. |
| **Invalid Or Duplicate** | Whenever the developers receive the defect and they do not approve that, it really is the defect and has the reason to believe that. The status will be changed to "Invalid Or Duplicate " |
| **Won’t Fix** | We also have a special team to check all the opened defects and decide or find the defect that is not real defect. The status will be changed to "Won’t Fix" |
| **Started** | The developer start to fix the defect. |
| **Fixed** | After the defect is assigned to developers, developers start to resolve the defect with provided solution from QA team. After the defect is resolved by developers, the status of defect will be changed to “Fixed”. |
| **Verified** | After the debeloper fix the defect. The tester will verify to ensure that the defects are fixed correctly. |
| **Done** | After the debeloper fix the defect. The tester will verify to ensure that the defects are fixed correctly,The status will be changed to "Done" |

**Table 8: Bug Priority Level**

|  |  |
| --- | --- |
| **Status** | **Description** |
| **Critical** | An item that prevents further testing of the product or function under test can be classified as Critical Bug. No workaround is possible for such bugs.   * Examples of this include a missing menu option or security permission required to access a function under test |
| **High** | A defect that does not function as expected / designed or cause other functionality to fail to meet requirements can be classified as Major Bug. The workaround can be provided for such bugs.   * Examples of this include inaccurate calculations; the wrong field being updated, etc. |
| **Medium** | The defects which do not conform to standards and conventions can be classified as Medium Bugs. Easy workarounds exists to achieve functionality objectives.   * Examples include matching visual and text links which lead to different end points. |
| **Low** | Cosmetic defects which does not affect the functionality of the system can be classified as Minor Bugs. |